

25. A semiconductor integrated circuit device,  
comprising:

B1 a memory array having a plurality of word lines, a plurality of bit lines, and a plurality of memory cells;

a processing circuit coupled to said memory array via a plurality of signal lines;

an input/output circuit coupled to one of the plurality of signal lines; and

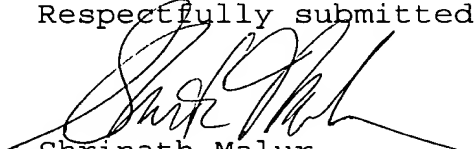
a switching circuit inserted between the plurality of signal lines and said input/output circuit. --

REMARKS

Claim 1 has been canceled. New claim 25 has been added. Accordingly, claim 25 is currently pending in the application.

Examination is respectfully requested.

Respectfully submitted,

  
Shrinath Malur  
Registration No. 34,663  
Attorney for Applicants

MATTINGLY, STANGER & MALUR  
104 East Hume Avenue  
Alexandria, Virginia 22301  
(703) 684-1120  
Date: December 20, 2000

MARKED UP  
SPEC

S P E C I F I C A T I O N

TITLE OF THE INVENTION

5        Neural Network Processing System Using Semicon-  
         ductor Memories

BACKGROUND OF THE INVENTION

10        The present invention relates to a data process-  
         ing system having a memory packaged therein for real-  
         izing a large-scale and fast parallel distributed pro-  
         cessing and, more specifically, <sup>for realizing</sup> [to] a neural network  
         processing system. ✓

15        ~~The~~ parallel distributed data processing using  
         ~~the~~ neural network, called the "neuro-computing" (as ✓  
         will be shortly referred to as the "neural network  
         processing") is noted in the field of acoustics,  
         speech and image processing, <sup>and is</sup> [as] described [in either] on ✓  
         pp. 145 - 168, "Parallel networks that learn to  
20        pronounce English text. Complex Systems 1 by Sej-  
         nowski, T. J., and Rosenberg, C. R. 1987, <sup>and</sup> [on] "Neural ✓  
         Network Processing" published by Sangyo Tosho and  
         edited by Hideki Asou. In [the] neural network process- ✓  
         ing, a number of processing elements, called [the] "neu-  
25        rons" <sup>are</sup> connected in a network <sup>and</sup> exchange [the] data through ✓  
         transfer lines called [the] "connections" for high-grade  
         data processing. In each neuron, the data (i.e., the  
         outputs of the neurons) sent from another neuron are  
         subjected to simple process[ing] such as multiplications ✓  
30        or summations. Since [the] processing in the individual ✓  
         neurons and [the] processing of different neurons can be ✓  
         carried out in parallel, the neural network processing  
         is advantageous in principle <sup>because it offers</sup> [in] its fast data process- ✓  
         ing. Since algorithms [or learnings] for setting the  
35        connection weights of the neurons for a desired data

processing have been proposed, [the] data processing can  
be varied for the objects, as described [in either] <sup>on</sup> pp. 533 - 536, "Learning representations by back-propagation errors.", Nature 323-9 (1986a) by Rumelhart, D.  
5 E., Hinton, G. E. and Williams, R. J., <sup>and on</sup> or in 2nd Section of "Neural Network Processing" published by Sangyo Tosho and edited by Hideki Asou.

#### SUMMARY OF THE INVENTION

10 First [of all], the operating principle of the neural network will be described in connection with two representative kinds: the multi-layered network and the Hopfield network. Fig. 2(a) shows the structure of the multi-layered network, and Fig. 3(a) shows the  
15 structure of the Hopfield network. Both of these networks are constructed <sup>using</sup> of the connections of neurons. Here, [are used] the terminology [of] "neurons", which will be called the <sup>neurons</sup> "nodes" or "processing elements", <sup>whatever</sup> [as] the case may be. The directions of the connection <sup>no</sup> [on] arrows indicate <sup>the direction in which the neuron outputs are transferred</sup> those of transferring the outputs of neurons.  
20 In the multi-layered network, as shown in Fig. 2(a), the neurons are stacked in multiple layers so that the neuron outputs are transmitted in the direction from the input to output layers only. Input signals  $IN_1$ , - - -, and  $IN_n$  are input[ed] to the input layer, and out-  
25 put signals  $OUT_1$ , - - -, and  $OUT_n$  are output[ed] from the output layer. In the Hopfield network, on the other hand, the neuron outputs are fed back to an identical neuron and are transferred in two <sup>directions</sup> [ways] be-  
30 tween <sup>two</sup> arbitrary [two] neurons. The feedback to the identical neuron may be omitted.

Figs. 2(b) and Fig. 3(b) show the processing principle to be accomplished in the neurons. This processing principle is similar in any network and  
35 will be described in connection with the multi-layered

network with reference to Fig. 2(b). Fig. 2(b) shows a j-th neuron in the (S+1)th layer in an enlarged scale. This neuron is fed through the connection with the output values  $V_{1s}$ , - - -,  $V_{is}$ , - - -, and  $V_{ns}$  of the neurons in the preceding layer, i.e., the S-th layer. Here, letters ns indicate the number of neurons in the S-th layer. In the neuron, the <sup>input</sup> products  $V_{1s}T_{1j}^s$ , - - -,  $V_{is}T_{ij}^s$ , - - -, and  $V_{ns}T_{nj}^s$  of the [inputted] output values  $V_{1s}$ , - - -,  $V_{is}$ , - - -, and  $V_{ns}$  of the neurons and the connection weights,  $T_{1j}^s$ , and so on, are calculated by means of a multiplier MT. Next, the sum of these products and an offset  $\Theta_{js+1}$  is calculated by means of an adder ADD. The offset  $\Theta_{js+1}$  may be omitted, as the case may be. Moreover, the result is inputted to a [circuit D] for nonlinear transfer function <sup>circuit D model</sup> to obtain the output value  $V_{js+1}$  of the neurons. The nonlinear transfer function circuit D has characteristics, as shown in Figs. 2(c) or 2(d), and outputs [an output]  $g(x)$  for an input  $x$ . Fig. 2(c) shows an example of the nonlinear transfer function [for outputting] <sup>which results in</sup> a binary output  $g_1$  or  $g_2$  <sup>depending</sup> [in dependence] upon whether or not the input  $x$  exceeds a predetermined threshold value  $x_{th}$ . [and] Fig. 2(d) shows an example using a sigmoid function for issuing continuous outputs. The nonlinear transfer function circuit D is given other characteristics, if necessary. As the case may be, on the other hand, the circuit D <sup>and even</sup> may be given linear characteristics.

The processing principle described above is also similar in the Hopfield network, as shown in Fig. 3(b). In the Hopfield network, however, not only the output of the neuron of the <sup>preceding</sup> layer, [preceding by one] but also the outputs of all neurons are inputted to one neuron. In the multi-layered network, as seen from Figs. 2(a) and 2(b), one processing is ended by <sup>first</sup> feed-

ing the output values of the neurons of the input layer [at first] and <sup>then</sup> by updating the output values of the neurons in the next layer and then by <sup>updating</sup> the <sup>outputs</sup> <sup>i.e. the secondary layer</sup> values of the neurons of the [secondary] next layer. In

5 the Hopfield network of Fig. 3(a), on the other hand, the output values of the individual neurons can be updated at suitable <sup>or</sup> timings because of <sup>the</sup> lack of any layer. In this Hopfield network, all the neuron output values are suitably given, and <sup>they are updated until</sup> their updating is continued till they come to an equilibrium state. <sup>in order to distinguish</sup> The net<sup>work</sup>, <sup>the one</sup> in which the output values of all neurons are simultaneously updated, is called the "synchronized Hopfield network", whereas the <sup>one</sup> network in which the output values are updated at arbitrary timings, 15 [are] called the "unsynchronized Hopfield network" [so that they are distinguished].

One method used for <sup>realizing above mentioned</sup> accomplishing the <sup>the</sup> aforementioned neural networks has employed [the] software whereas the other <sup>has employed</sup> the hardware. According to the 20 method employing [the] software, the processing of <sup>the</sup> neurons is carried out with a program written in computer languages so that the number or structure of the neurons can be easily changed. Since, however, the processing is sequentially performed, the former 25 method is disadvantageous in that the data processing time is abruptly elongated for an increased number of neurons. In the Hopfield network using an  $n$  number of neurons, an  $n$  times of products have to be calculated for [updating] <sup>to update</sup> the output of one neuron. In order to 30 update the output values of all neurons at least once, therefore, an  $n^2$  times of products [need to] <sup>must</sup> be calculated. In other words, the number of calculations will increase [in] <sup>an</sup> the order of  $n^2$  with [the] <sup>an</sup> increase in the neuron number  $n$ . As a result, the data processing 35 time will increase [in] <sup>an</sup> the order of  $n^2$  if [the] multi-

plications are sequentially accomplished.

According to the method employing <sup>putting</sup> the hardware, the processing time can be shortened by changing the neurons to be multiplied into the hardware. Another trial for speeding up the processing has been made by executing the processing in parallel with a number of hardware neurons. If, however, the number of neurons is enlarged, the number of wiring lines acting as the signal lines between the neurons will increase <sup>on</sup> in the order of  $n^2$ , thus making it difficult to realize a large-scale network.

The method of solving the wiring problem is exemplified on pp. 123 - 129 of Nikkei Microdevice, March, 1989, <sup>and</sup> as will be described in principle in Fig. 4. Fig. 4 shows an example, in which a multi-layered network <sup>is</sup> composed of three layers, each having three neurons <sup>and</sup> is constructed of analog neuro-processors ANP and SRAM. The ANP is made by integrating one multiplier <sup>MT</sup>  $M$  and one adder ADD of Fig. 2(b) and a non-linear transfer function circuit D <sup>on to</sup> into one chip. Another chip SRAM is stored with the connection weight belonging to each neuron. The neurons of different layers are connected through one signal line called the "analog common bus". Since the neuron output value of an input layer is inputted from the outside, the ANP and SRAM corresponding to the neurons of the input layer are omitted.

The operations are as follows. First <sup>necessary for the desired data processing</sup> [of all], the connection weight <sup>necessary for a desired data processing</sup> for each ANP is read in each SRAM from the outside. Next, an input analog signal corresponding to the output value of one neuron in the input layer is inputted. The input signals are inputted in parallel directly to the ANPs corresponding to the neurons of the middle layer. The weight data are read out from the

SRAM to each ANP in synchronism with the input signal.

Next, the product of two signals is calculated, and the result is stored in each ANP. Next, the input

analog signals corresponding to the output values of

5 other neurons of the input layer are inputted, and

their product is calculated. The calculation result

is added to the value which is stored in each ANP of

the middle layer. After similar calculations have

been accomplished for the input analog signal cor-

10 responding to the output value of the last neuron of

the input layer, the neuron output values  $V_{12}$ ,  $V_{22}$  and

$V_{32}$  of the middle layer are consecutively outputted to

the analog common bus of the middle layer by the non-

linear transfer function circuit in the ANP so that

15 <sup>similar</sup> calculations similar the foregoing ones are continued.

Finally, the neuron output values  $V_{13}$ ,  $V_{23}$  and  $V_{33}$  of

the output layer are consecutively outputted to the

analog common bus of the output layer by the nonlinear

transfer function circuit in the ANP of the output

20 layer.

According to the example of the prior art de-

scribed above with reference to Fig. 4, the wiring

problem can be avoided by driving the common bus in

the time sharing manner. Since, moreover, the multi-

25 plications of the number of the neurons in one layer

can be carried out in parallel, the data processing

rate can be drastically raised, as compared with the

method <sup>utilizing</sup> according to the <sup>software</sup> software, <sup>since hardware processing is faster</sup> as a result of the

speed-up of the processing by the hardware.

30 Since, however, the ANPs and the SRAMs are <sup>on</sup> formed

in different chips, it is difficult to realize <sup>a high density,</sup> the

large-scale network in high density. Even if thirty

two neurons could be integrated over a square chip of

10 mm, as described on pp. 129 of the above-specified

35 Nikkei Micro Device, March, 1989, one adder, one mul-

tiplier and one nonlinear transfer function circuit need<sup>to</sup> be prepared for each neuron. Hence, it is difficult to integrate several hundreds or thousands<sup>of</sup> neurons ~~over~~<sup>on</sup> one chip. ✓✓

5        Moreover, the examples of the prior art described above ~~is~~<sup>are</sup> accompanied by the following problems when they are applied to practices. An application of the multi-layered network is exemplified by ~~outputting~~<sup>one</sup> ~~the~~<sup>of</sup> the pronunciations and accents of English letters input~~ted~~ ✓  
10       to a three-layered network, as described on pp. 145 - 168 of Complex Systems 1 (1987) entitled "Parallel networks that learn to pronounce English text" and edited by Sejnowski, T. J., and Rosenberg, C. R. Seven English letters are encoded as the neuron output values of the first layer, and the codes corresponding to the pronunciations and accents of the central one of the aforementioned seven letters are output~~ted~~ as ✓  
15       the output values of the twenty six neurons of the third layer. In this example, ~~in dependence~~<sup>depending</sup> upon the inputs, the output values of the neurons of the output layer may fail to coincide with the codes of the pre-defined pronunciations and accents but may give fuzzy values. It is, therefore, necessary, to find ~~out~~ the ✓  
20       codes closest to those of all the pronunciations and accents<sup>and</sup> ~~compared~~<sup>and</sup> ~~to make~~<sup>them</sup> ~~then~~<sup>them</sup> ~~as~~<sup>as</sup> the correct answer. These comparisons between the output values and the desired values (i.e., the codes of the pronunciations and accents in the aforementioned example) of the outputs are likewise required for ~~the~~ pattern ✓  
25       recognition by the neural network. This point is not taken into consideration in the aforementioned examples of the prior art so that the examples are ~~in-~~<sup>not applicable</sup> convenient when applied to the practical problem. ✓✓  
30       In the ~~aforementioned~~<sup>above mentioned</sup> examples of the prior art, ✓

35       moreover, the connection weight necessary for the



desired data processing is determined by an external computer, and the result is written <sup>into</sup> [in] the SRAMs of Fig. 4. As a result, the learning is wholly performed by the software <sup>which makes</sup> [so that] it [is] difficult to carry out at [a] high speed.

In order to solve the problems thus far described, according to the present invention, there is provided a neural network processing system which comprises: a memory for storing neuron output values, connection weights, the desired values of outputs, and data necessary for learning; an input/output circuit for writing or reading data in or out of said memory; an arithmetic circuit for performing a processing for determining the neuron outputs such as the product, sum and nonlinear conversion of the data stored in said memory, a comparison of the output value and its desired value, and a processing necessary for learning; and a control circuit for controlling the operations of said memory, said input/output circuit and said arithmetic circuit. The arithmetic circuit is constructed to include at least one [of an] adder, a multiplier, a nonlinear transfer function circuit and a comparator so that at least a portion of the processing necessary for determining the neuron output values such as the product or sum may be accomplished in parallel. Moreover, these circuits are shared among a plurality of neurons and are operated in a time sharing manner to determine the plural<sup>ty</sup> of neuron output values. Still, moreover, the <sup>above</sup> [afore]mentioned comparator compares the neuron output value determined and the desired value of the output in parallel.

Since the adder, the multiplier and the nonlinear transfer function circuit are shared among the plural<sup>ty</sup> of neurons, as described above, the system for calculating the neuron output values in the neural network

using [the] numerous neurons can be realized by using a  
small number of circuits. Since, moreover, at least a  
portion of the neuron processing, such as the product  
and sum, is performed in parallel by the aforementioned  
5 arithmetic circuit, the data processing can be ac-  
complished at [a] high speed. [Since, still moreover,<sup>Furthermore, since</sup>  
the output value obtained and the desired value of the  
output can be compared in parallel by the aforemen-  
tioned comparator, the distance (i.e., the similarity  
10 between the obtained output value and the expected  
value, e.g., the <sup>hamming</sup> [distance] between the ob-  
tained output value and the desired value can be  
determined at [the] high speed. Since, furthermore, at  
least a portion of the processing necessary for the  
15 learning is accomplished by the hardware of the sys-  
tem, the learning<sup>speed</sup> can be <sup>increased</sup> [speeded up].

An object of the present invention is to solve  
the problems owned by the aforementioned prior art.

Another object of the present invention is to  
20 provide a system for carrying out the calculations of <sup>the</sup>  
neuron output values at [a] high speed with a small num-  
ber of circuits in the network containing a number of  
neurons.

Still another object of the present invention is  
25 to give the aforementioned system a function to com-  
pare the neuron output value and the desired value at  
[a] high speed.

A further object of the present invention is to  
give the aforementioned system a function to process  
30 at least a portion of the processing necessary for the  
learning.

These and other objects and many of the attendant  
advantages of the present invention will be readily  
appreciated as the same becomes better understood by  
35 reference to the following detailed description when

considered in connection with the accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

5        Fig. 1 shows one embodiment of the structure, in which the data processing system according to the present invention is embodied over a single semiconductor chip;

10       Figs. 2(a) and 2(b) are diagrams showing the principle of the multi-layered neural network;

      Figs. 2(c) and 2(d) are diagrams showing the examples of the characteristics of the nonlinear transfer function circuit D;

15       Figs. 3(a) and 3(b) are diagrams showing the principle of the Hopfield neural network;

      Fig. 4 shows an example of the neural network processing system using the plural chips according to the prior art;

20       Figs. 5(a) and 5(b) show embodiments of the method for parallel processing of the multi-layered neural network and the Hopfield neural network, respectively;

25       Figs. 6(a) and 6(b) show second embodiments of the method for parallel processing of the multi-layered neural network and the Hopfield neural network, respectively;

30       Fig. 7(a) shows one embodiment of the structure, in which the data processing system according to the present invention is realized by using a memory array capable of reading out a number of data on data lines by selecting one word line;

35       Fig. 7(b) shows one embodiment of the structure, in which the data processing system according to the present invention is realized by using two memory arrays capable of reading out a number of data on data

lines by selecting one word line;

Fig. 8(a) shows one embodiment of the corresponding relations of memory cells to the neuron output values and the connection weights, in <sup>the</sup> case <sup>where</sup> the multi-layered neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output value and the connection weight are realized with each memory cell;

Fig. 8(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit D which is suitable in <sup>the</sup> case <sup>where</sup> binary memory cells are <sup>to be</sup> used in the embodiment of Fig. 8(a);

Fig. 8(c) shows one embodiment of the characteristics of the nonlinear transfer function circuit D which is suitable in <sup>the</sup> case <sup>where</sup> tetral memory cells are used in the embodiment of Fig. 8(a);

Fig. 8(d) shows one embodiment of the method of selecting the word lines and the memory cells in the arithmetic mode in the embodiment of Fig. 8(a);

Fig. 9(a) shows a second embodiment of the corresponding relations of the memory cells to the neuron output values and the connection weights, in <sup>the</sup> case <sup>where</sup> the multi-layered neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output values and the connection weights are realized <sup>using</sup> by a plurality of memory cells;

Fig. 9(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit D which is suitable in <sup>the</sup> case <sup>where</sup> the neuron output values and the connection weights are equivalently expressed by a plurality of memory cells in the embodiment of Fig. 9(a);

Fig. 9(c) shows one embodiment of a method of setting the values  $X_{th1}$ , - - -, and  $X_{thp}$  of Fig. 9(b)

for giving the nonlinear transfer function circuit D the sigmoid characteristics;

Fig. 10(a) shows a third embodiment of the corresponding relations of the memory cells to the neuron output values and the connection weights, in <sup>the</sup> case <sup>where</sup> the multi-layered neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output values and the connection weights are realized <sup>using</sup> by a plurality of memory cells;

Fig. 10(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit D which is suitable in <sup>the</sup> case <sup>where</sup> the neuron output values and the connection weights are binarily expressed by a plurality of memory cells in the embodiment of Fig. 10(a);

Fig. 11 shows one embodiment in <sup>the</sup> case <sup>where</sup> a DRAM cell is used as the memory cell of the embodiment of Fig. 8(a);

Fig. 12 shows one embodiment of the relations between the arithmetic modes and the external signals;

Fig. 13(a) shows one embodiment of the operation waveforms in the memory mode of the embodiment of Fig. 11;

Fig. 13(b) shows one embodiment of the operation waveforms in the arithmetic mode of the embodiment of Fig. 11;

Fig. 14(a) shows one embodiment of the nonlinear transfer function circuit D which is suitable for the embodiment of Fig. 11 or the like;

Fig. 14(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit D of Fig. 14(a);

Fig. 14(c) shows one embodiment of the input/output circuit IO which <sup>is</sup> suitable for the embodiment of

Fig. 11 or the like;

Fig. 14(d) shows one embodiment of the potential relation between the product and sum output line NO and the write line IA in <sup>the</sup> case <sup>where</sup> the nonlinear transfer function circuit of Fig. 14(a) and the write circuit of Fig. 14(c) are used; ✓

Fig. 14(e) shows one embodiment of the read circuit OUT which is suitable for the embodiment of Fig. 11 or the like;

Fig. 14(f) shows one embodiment of the read latch circuit OUTLT which is suitable for the embodiment of the read circuit OUT of Fig. 14(e);

Fig. 15 shows a second embodiment of the multiplier MT in Fig. 11;

Figs. 16(a) and 16(b) show examples of the SRAM cell;

Fig. 16(c) shows one embodiment in <sup>the</sup> case <sup>where</sup> the SRAM cells of Figs. 16(a) and 16(b) are used as the memory cell of Fig. 8(a); ✓

Fig. 17(a) shows one embodiment of the operation waveforms in the memory mode of the embodiment of Fig. 16(c);

Fig. 17(b) shows one embodiment of the operation waveforms in the arithmetic mode of the embodiment of Fig. 16(c);

Fig. 18(a) shows one embodiment in <sup>the</sup> case <sup>where</sup> the DRAM cells are used in the embodiment of Fig. 9(a) or the embodiment of Fig. 10(a); ✓

Fig. 18(b) shows one embodiment of the structure of the block BLK1 in the embodiment of Fig. 18(a);

Fig. 18(c) shows one embodiment of the structure of the block BLK2 and the nonlinear transfer function circuit D in the embodiment of Fig. 18(a);

Fig. 18(d) shows one embodiment of the structure of the individual nonlinear transfer function circuits

DSx ( $x = 1, 2, \dots$ , and  $p$ ) composing the nonlinear transfer function circuit D10 in the embodiment of Fig. 18(c);

5 Fig. 18(e) shows one embodiment of the characteristics of the individual nonlinear transfer function circuit DSx ( $x = 1, 2, \dots$ , and  $p$ ) composing the nonlinear transfer function circuit D10 in the embodiment of Fig. 18(c);

10 Fig. 19(a) shows one embodiment of the structure of the nonlinear transfer function circuit D10 which is suitable in case the neuron output values and the connection weights are binarily expressed in a plurality of memory cells in the embodiment of Fig. 18(a);

15 Fig. 19(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit DSx ( $x = 1, 2, \dots$ , and  $z$ ) in the embodiment of Fig. 18(a);

20 Fig. 19(c) shows one embodiment of the characteristics in the embodiment of Fig. 19(a);

Fig. 19(d) shows one embodiment of the structure of the encoder in the embodiment of Fig. 19(a);

25 Fig. 20(a) shows one embodiment of the corresponding relations of memory cells to the neuron output values and the connection weights, in case the unsynchronized Hopfield neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output value and the connection weight are realized [with] each memory cell; ✓

30 Fig. 20(b) shows one embodiment of the corresponding relations of memory cells to the neuron output values and the connection weights, in case the synchronized Hopfield neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output value and the con-  
35

nection weight are realized <sup>in</sup> [with] each memory cell; ✓

Fig. 21(a) shows one embodiment of the corresponding relations of memory cells to the neuron output values and the connection weights, in <sup>the</sup> case <sup>where</sup> the unsyn- ✓  
5 chronized Hopfield neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output value and the connection weight are realized with a plurality of memory cells;

10 Fig. 21(b) shows one embodiment of the corresponding relations of memory cells to the neuron output values and the connection weights, in <sup>the</sup> case <sup>where</sup> the syn- ✓  
15 chronized Hopfield neural network is realized by using the embodiment of Fig. 7(b), that is to say, the embodiment in which the neuron output value and the connection weight are realized with a plurality of memory cells;

Fig. 22 shows one embodiment in <sup>the</sup> case <sup>where</sup> the neuron ✓  
output values and the connection weights are enabled  
20 to take positive and negative values by using coding bits;

Fig. 23 shows one embodiment in <sup>the</sup> case <sup>where</sup> the system ✓  
according to the present invention is given a function to compare the neuron output values and the desired  
25 values;

Fig. 24 shows one embodiment of the comparator for comparing the data read out to a plurality of data line pairs of the memory cell array TG and the memory cell array A to calculate the extent of similarity of  
30 the data; and

Fig. 25 shows one embodiment in which [the updating of] the neuron output values <sup>are updated faster</sup> [is speeded up] by providing a register. ✓ ✓

35 DETAILED DESCRIPTION OF THE INVENTION



Fig. 1 shows one embodiment in <sup>the</sup> case <sup>the</sup> the data processing system according to the present invention is integrated over a semiconductor chip.

There <sup>items</sup> are <sup>include</sup> integrated over a semiconductor chip (CHIP): a memory (A) for storing data; an input/output circuit (I/O) for performing at least one of the writing operation and reading the data in and from said memory; an arithmetic circuit (ARTM) for performing the arithmetic for determining neuron output values, the comparison (i.e., the similarity of the obtained output values and the desired values, e.g., the calculation of the <sup>humming</sup> distance) of the output values and the desired values or the arithmetic necessary for <sup>the</sup> learning by using the data stored in said memory; and a control circuit (CNT) for controlling the operations of said memory, said input/output circuit and said arithmetic circuit. Each of <sup>the</sup> buses (BUS1, BUS2, BUS3, BUS4, etc.) connecting the individual blocks is made of not only one wiring line but also a necessary number of wiring lines. The aforementioned memory can be stored with the connection weights and the neuron output values necessary for <sup>the</sup> neural network processing, the desired values of the outputs or the data necessary for the learning. According to the present embodiment, <sup>the</sup> nonlinear network processing such as the calculations of the neuron output values, the comparisons of the <sup>desired</sup> <sup>output</sup> values <sup>and</sup> <sup>the</sup> desired values, or the calculations necessary for <sup>the</sup> learning can be performed in the following <sup>ways</sup> <sup>manner</sup>s.

First <sup>of all</sup>, the method of calculating the neuron output values will be described <sup>in the following</sup>. <sup>At</sup> first, the connection weights necessary for the calculations for <sup>the</sup> neural network processing, and the neuron output values, or the offsets are read out in parallel from the memory to the arithmetic circuit

through a bus 1. Next, the <sup>calculations</sup> arithmetics such as the product and sum or the nonlinear transfer necessary for determining the neuron output values are accomplished by the arithmetic circuit, and the obtained results are written in the memory through the input/output circuit. The operations described above are continued by <sup>the</sup> a necessary number of times to determine the neuron output values. The arithmetic circuit may either determine one of <sup>a</sup> plural <sup>by</sup> neuron output values by a single operation or perform a portion of calculations for determining the neuron output values. Thus, the data processing can be accomplished by the various networks such as the multi-layered network or the synchronized or unsynchronized Hopfield network. Incidentally, in order to update the output values of all the neurons synchronously, the synchronized Hopfield network needs to be stored with the output values of all the neurons <sup>until all</sup> [till] the <sup>have been updated</sup> end of updating the output values of all the neurons. In this case, the output values of all the previous neurons may be stored in the memory so that they may be used for updating the output values of the neurons.

According to the present embodiment, a desired number of multipliers, adders and nonlinear transfer function circuits necessary for calculating the neuron output values may be provided in the arithmetic circuit so that they may be <sup>used</sup> repeatedly [used]. This makes it possible to make the number of circuits far smaller than [that of] the case in which those circuits are prepared for each of the neurons. The example of the prior art of Fig. 4 [is] required <sup>5</sup> [to prepare] two hundreds multipliers, adders and nonlinear transfer function circuits for realizing the multi-layered network having three layers each composed of one hundred neurons. In the present embodiment, on the contrary, it

is sufficient to prepare at least one multiplier, at least one adder and at least one nonlinear transfer function circuit. Even if the multiplications necessary for updating one neuron output value <sup>would</sup> to speed up the operations should be accomplished in parallel, it would be sufficient to prepare one hundred multipliers, one adder and one nonlinear transfer function circuit. According to the present embodiment, therefore, the number of circuits can be drastically reduced, as compared with that of the prior art. Incidentally, the above-specified <sup>difference</sup> difference will become <sup>the</sup> larger for <sup>the</sup> larger scale of <sup>the</sup> network. Similar situations will apply to another network such as the Hopfield network.

Not only the <sup>calculation</sup> arithmetic speed such as multiplications, but also the amount of <sup>calculations</sup> arithmetics to be carried out in parallel makes <sup>a significant</sup> high contribution to the data processing speed <sup>where</sup> in case the neuron output values are to be determined. In the Hopfield network using an n number of neurons, for example, <sup>n<sup>2</sup></sup> the products of n<sup>2</sup> times have <sup>must</sup> to be calculated for updating the output values of all the neurons, as has been described <sup>earlier</sup> hereinbefore. If the multiplications <sup>accomplished</sup> are sequentially accomplished, therefore, the updating of the output values of all the neurons takes at least a time period of n<sup>2</sup> times as long as that required for one multiplication. As a result, the time period required for the multiplications will abruptly increase <sup>in</sup> the order of <sup>the</sup> square of the neuron number with <sup>the</sup> increase in the number of neurons. Similar circumstances will also apply to the multi-layered network. This makes it desirable to calculate <sup>multiple</sup> the numerous multiplications in parallel. <sup>next</sup> [Here will be described in the following] an example of the arithmetic system for raising the data processing speed by making the multiplications in

parallel so as to determine the neuron output values  
in the embodiment of Fig. 1. <sup>will be described</sup> ✓

Fig. 5 illustrates the multi-layered network at  
(a) and the Hopfield network at (b) of one system for  
5 [the] parallel <sup>computations</sup> [arithmetics]. In the present embodiment, ✓  
the products necessary for determining one neuron out-  
put value are calculated, as shown. Specifically, the  
output value of the neuron of the preceding layer,  
which is input<sup>ted</sup> to one neuron, and the connection ✓  
10 weight for <sup>the</sup> said output value of the neuron under con- ✓  
sideration are read out in parallel from the memory,  
and their products are calculated in parallel. Thus,  
the time period required for the multiplications will  
increase <sup>in</sup> the order of the neuron number with [the] <sup>an</sup> ✓  
15 increase in the neuron number. As a result, the data  
processing time can be drastically shortened, as com-  
pared with the case in which the multiplications are  
<sup>accomplished</sup> sequentially [accomplished]. In Fig. 5, only the multi- ✓  
plications necessary for updating the output value of  
20 one neuron are <sup>executed</sup> [accomplished] in parallel. However, the ✓  
embodiment of Fig. 1 should not be limited thereto but  
may naturally add the arithmetic circuits within a  
range allowed by the degree of integration, to update  
the output values of the plural <sup>ty of</sup> neurons in parallel. ✓  
25 In this case, [the] data processing can be accomplished  
at [a] <sup>higher</sup> speed. In addition, the parallel [arith-  
metics] <sup>calculations</sup> can also be accomplished by another system, as ✓  
shown in Figs. 6(a) and 6(b). ✓

Fig. 6 shows one embodiment, in which the multi-  
30 plications are executed in parallel for a plurality of  
neurons to be fed with the output value of one neuron  
in the multi-layered network of Fig. 6(a) and in the  
Hopfield network of Fig. 6(b). In this method, the  
neural output values and the connection weights are  
35 read out from the memory, and the calculations neces-

sary for updating the neuron output values are executed bit by bit for the plural<sup>ity of</sup> neurons. This makes it impossible to realize the unsynchronized Hopfield network. Since, however, the time period required for the multiplications will increase  $[in]^{m}$  the order of the neuron number with the increase in the neuron number like the system of Fig. 5, the data processing time can be drastically shortened, as compared with the case in which the multiplications are sequentially carried out.

In the example of the prior art of Fig. 4, too, the arithmetic<sup>calculations</sup> are executed in parallel. As will be described <sup>below</sup> in the following, however, the structure of Fig. 1 can be realized with a smaller number of circuits than that of the example  $[of]^{m}$  the prior art. In the systems shown in Figs. 6(a) and 6(b), only one multiplier operates in parallel in each neuron, as hatched. In the embodiment of Fig. 1, therefore, the arithmetic circuit may be provided with the multipliers in a number equal to that of the neurons to be calculated at one time. <sup>this is done</sup> so that this system can be realized with a smaller number of circuits than that  $[of]^{m}$  the case of the prior art in which all the multipliers are provided for all the individual neurons. In the multi-layered network having three layers, each composed of three neurons, for example, a similar parallelism can be realized by using individually three <sup>individual</sup> multipliers, adders and nonlinear transfer function circuits for example, according to the embodiment of Fig. 1. <sup>this is</sup> on the contrary to the case of the prior art which is equipped with individually <sup>individual</sup> six multipliers, adders and nonlinear transfer function circuits.

Thus, according to the embodiment shown in Fig. 1, a system for the data processing similar to that of the neural network using numerous neurons can be real-

ized with the <sup>minimum number of</sup> necessary minimum circuits by sharing  
the adders, multipliers and nonlinear transfer func-  
tion circuits of the arithmetic circuit among the  
plural <sup>ity of</sup> neurons. By executing the arithmetics <sup>calculations</sup> such as  
5 the products or sums with the aforementioned arith-  
metic circuit, moreover, the data processing can be  
accomplished at [a] high speed. Incidentally, with the <sup>calculations</sup>  
parallel arithmetics, the number of wiring lines be-  
10 tween the memory and the arithmetic circuit has to be  
increased to send many data at once to the arithmetic  
circuit. In Fig. 1, however, the memories and the  
arithmetic devices are arranged over the <sup>a</sup> common chip  
so that the number of the wiring lines on <sup>m</sup> the bus can  
be easily increased.

15 Although the method of calculating the neuron  
output values has been described hereinbefore, a neu-  
ron output value and its desired value can be compared  
according to the embodiment of Fig. 1. For this com-  
parison, the desired value may be stored in advance in  
20 the memory so that its distance from the output value  
obtained by the aforementioned method may be calcu-  
lated by the arithmetic circuit. This operation is to  
calculate the similarity between the desired value and  
the calculated value. At this time, the desired value  
25 and the output value, composed of numerous bits, can be  
simultaneously read out to the arithmetic circuit and  
processed in parallel with ease by increasing the num-  
ber of wiring lines on <sup>m</sup> the bus 1. Thus, according to  
the embodiment of Fig. 1, the data processing such as  
30 the pattern recognition can be executed at [a] high  
speed, as compared with the case in which the com-  
parison is accomplished serially bit by bit by using  
an external computer.

35 According to the embodiment of Fig. 1, moreover,  
the learning can be accomplished at [a] higher speed

than that of the case using [the] software, by executing  
the [arithmetics] <sup>calculations</sup> necessary for [the] learning with the  
arithmetic circuit. This specific embodiment will be  
described <sup>below</sup> [hereinafter].

5 [The] <sup>N</sup>neural network processing is advantageous in  
that it can process various data by changing the con-  
nection weights. This advantage can be easily ex-  
ploited according to the embodiment of Fig. 1 by re-  
writing the connection weight stored in the memory.

10 Moreover, several kinds of connection weights neces-  
sary for different data processings can be stored in  
advance by making the capacity of the memory larger  
than that necessary for calculating the neuron output  
values. In this case, [there can be attained a merit]

15 <sup>The benefit</sup> that different kinds of data can be continuously  
processed without losing the time period for rewriting  
the connection weights. <sup>can be realized</sup> In addition, in <sup>the</sup> case <sup>where numerical</sup> [numerical]  
input data are to be continuously processed, the nec-  
essary input data or the obtained data can be stored  
20 in advance in a portion of the memory. Thus, the  
frequency for switching the reading, calculating and  
output[ing] modes can be reduced to shorten the pro-  
cessing time, as compared with the case in which the  
operations of reading each input data in the memory  
25 and calculating and outputting it <sup>externally</sup> [to the outside of  
the system] are repeated.

Next, a more specific embodiment, based upon the  
embodiment of Fig. 1, will be described [in the follow-  
ing]. For simplicity [of descriptions], the case <sup>where</sup> [of  
30 giving] the arithmetic circuit <sup>is given</sup> the function of cal-  
culating the neuron output values will be described at  
first, and the method of giving the comparing or  
learning function will be described <sup>later</sup> [at last].

Fig. 7(a) shows one embodiment in <sup>the</sup> case <sup>where</sup> a lattice-  
35 shaped memory cell array is used in the memory of the

embodiment of Fig. 1. In Fig. 7(a), letter A designates a memory cell array which is composed of: a plurality of data lines (D); a plurality of word lines (W) arrayed to intersect the data lines (D); and memory cells (MC) arrayed at the desired intersections.

As a result, the signals of the different memory cells can be read out onto the plural<sup>ity of</sup> data lines by selecting one of the word lines. Numeral 12 designates an arithmetic circuit (ARTM). Numerals 10, 11, 13, 14,

15 and 16 designate circuits corresponding to the control circuit (CNT) of Fig. 1. The numerals 10 and 15 designate address buffers for X-addresses and Y-addresses, and the numerals 11 and 14 designate a decoder and a driver for X-addresses and Y-addresses, respectively. [The numeral 13 designates an array control circuit for controlling the memory cell array.

Numeral 16 designates a clock generator for generating clocks to control[s] the operations of the memories on the basis of the signals input[ted] from the outside.

Letters OUT and WR designate a read circuit and write circuit, respectively. A chip select  $\overline{CS}$  is a chip selection signal. A write control signal  $\overline{WE}$  is a signal for switching the write and read operations for establishing the write operation at a low level and the read operation at a high level. Letter  $\overline{NE}$  designate an arithmetic circuit control signal for starting the arithmetic circuit at a low level and interrupting the same at a high level to act as an ordinary memory.

In the following, the state of the signal  $\overline{NE}$  at the high level will be called the "memory mode", and the state at the low level will be called the "arithmetic mode". In the memory mode, a desired memory cell is selected according to the X-address and the Y-address so that a write data DI can be written in that cell or so that (a) data can be read out from the same cell and



outputted] as a read data DO. In the arithmetic mode, ✓  
the data stored in the memory cell is read out to the  
arithmetic circuit 12 so that the arithmetic result of  
the arithmetic circuit 12 or the data according to the  
5 arithmetic result can be written in the memory cell  
through the input circuit. By selecting one word  
line, according to the present embodiment, the data of  
all the memory cells on the selected word are out-  
putted] to the data lines. As a result, numerous data ✓  
10 can be easily latched in the arithmetic circuit 12 so  
that many [arithmetics] <sup>calculations</sup> can be accomplished in parallel. ✓  
In order to calculate the neuron output values accord-  
ing to the present embodiment, the mode <sup>first set</sup> is [changed at  
first] [into the memory mode to stop the arithmetic cir- ✓  
15 cuit, and the necessary connection weight, neuron out-  
put value (i.e., the input signal at first), offset  
and so on are written in the memory. Next, the mode  
is [changed] <sup>set</sup> to the arithmetic mode to start the arith- ✓  
metic circuit [to] read <sup>ing</sup> the necessary data [is read] out ✓  
20 to the arithmetic circuit by selecting one word line.  
Next, the result is written in the memory circuit. If.  
the read of the data necessary for the [arithmetics] <sup>calculations</sup> and  
the write of the result are further continued [by] [a] ✓  
the necessary number of times, [the] neural network process- ✓  
25 ing can be accomplished at [a] high speed. As has been ✓  
described above, according to the embodiment shown in  
Fig. 7(a), many data can be written at once in the  
arithmetic circuit [the] <sup>mode</sup> the embodiment is suited for [the] ✓  
parallel [arithmetics] <sup>calculations</sup> of the type shown in Fig. 5 or ✓  
30 Fig. 6. Thus, according to the present embodiment,  
[the] parallel <sup>calculations</sup> [arithmetics] make it possible to execute ✓  
[the] neural network processing at [a] high speed. By ✓  
using the arithmetic circuit 12 repeatedly, moreover,  
the plural <sup>th of</sup> neurons can share the output value <sup>with the</sup> cal- ✓  
35 culating circuit to effect a high <sup>degree of</sup> integration [easily]. ✓

In<sup>the</sup> case, on the other hand, <sup>where</sup> parallel <sup>calculations</sup> arithmetics are to be accomplished by using the data stored in the memory cells on the plural <sup>word lines</sup>, <sup>primary storage</sup> a register [for primary storage] can be provided in the arithmetic circuit so that it may [once] store <sup>one time</sup> the data obtained by selecting the word lines and execute the <sup>calculations</sup> [arithmetics] of the stored data together with the data read out by selecting other word lines.

As in the embodiment shown in Fig. 7(b), moreover, two memories, A and B can be provided. In Fig. 7(b), characters 13A and 13B designate array control circuits for controlling the memory cell arrays A and B, respectively. Other circuits such as a decoder is not shown in Fig. 7(b). According to the structure of Fig. 7(b), the [data of the] memory cells <sup>data</sup> on the two word lines of [the] memory cell arrays A and B can be written in the arithmetic circuit by selecting one word line [of] each of the memory cell arrays A and B. By using the structure of Fig. 7(b), <sup>because</sup> the memory arrays can be <sup>used</sup> separately [used] according to the kinds of data such that [the] memory cell array A [is] <sup>can</sup> stored [with] the neuron output value whereas [the] memory cell array B <sup>can</sup> [is] stored [with] the connection weight, the controls of the reading or writing operations can be simplified. Incidentally, in the embodiments of Figs. 7(a) and 7(b), the write data DI and the read data DO may be processed in plurality and in parallel, or [the] array<sup>s</sup> A and B may be <sup>provided</sup> separately [provided] with the read circuit OUT and the write circuit WR.

In the embodiments of Figs. 7(a) and 7(b), the selection of a specific memory cell can be accomplished like the ordinary memory according to the address. By changing the order of selecting the address, therefore, those embodiments can be flexibly applied to the various networks or various parallel

arithmetic systems.

In the embodiments of Figs. 7(a) and 7(b), the memory can be exemplified by a highly integrated semiconductor memory such as the DRAM or SRAM. In this case, the memory can store many data so that a large-scale network can be integrated into one chip.

Next, the method of realizing the multi-layered network by using the structure of Fig. 7(b) will be described <sup>below</sup> in detail [in the following]. The parallel arithmetic system is exemplified by taking the system of Fig. 5(a). It is assumed that the number of layers be  $m$  and that the number of neurons in each layer be  $n$ . Incidentally, the offset  $\Theta$  of each neuron, as shown in Figs. 2(b) or Fig. 3(b), will be omitted here

[so as] to simplify the description. As is apparent from Fig. 2(b) or Fig. 3(b), however, the offset  $\Theta$  of each neuron can be handled like the output from another ordinary neuron. <sup>This may be accomplished by two methods.</sup> <sup>The first provides</sup> either by providing one neuron having an output value of 1 at all times to setting the connection weight of it and each neuron  $[at]$  <sup>to</sup> the offset  $\Theta$ . <sup>The second method increases</sup> or by increasing the neuron output value, which is to be inputted <sup>in order</sup> for each neuron, by 1 <sup>to</sup> set the value  $[at]$  <sup>to</sup> the offset  $\Theta$  of each neuron and the corresponding connection weight  $[at]$  <sup>to</sup> 1 so that their product may be added to the total sum of the products of other neuron output values and the connection weights.

Fig. 8(a) shows one embodiment in which the memory cells [are] correspond to the connection weights and the neuron output values. Letter D designates the non-linear transfer function circuit; characters  $c_1, c_2, -$  -, and  $c_n$  designate the adders; and characters  $m_1, m_2, -$  -, and  $m_n$  designate the multipliers. The adders  $c_1, c_2, -$  -,  $c_n$  <sup>together</sup> constitute [altogether] the multi-input adder ADD of Fig. 2(b). The memory cell array A is stored with the neuron output values, and

the memory cell array B is stored with the connection weights. Although what is shown in Fig. 8(a) is the memory cells for storing the neuron output values and the connection weights, it is quite natural that the memory cells <sup>may</sup> be stored with other data such as the offsets  $\theta$  of the individual neurons or the data necessary for the learning [may be provided], if necessary. ✓  
As shown, the memory cells located at the intersections of the word lines  $s$  and the data lines  $i$  in the memory cell array A are stored with neuron output values  $V_{is}$ . In other words, the output values of the neurons of the common layer are arranged on the common word line. In the memory cell array B, the memory cells located at the intersections between the word lines  $(s, j)$  and the data lines  $i$  are stored with connection weights  $T_{ij}^s$ .

Figs. 8(b) and 8(c) show one embodiment of the input/output characteristics of the nonlinear transfer function circuit D. Fig. 8(b) shows the embodiment having binary outputs  $g1$  and  $g2$ . Characters  $x1$  and  $x2$  indicate the lower limit and the upper limit of the input  $x$ , respectively. In Fig. 8(b), the output is  $g2$ , if the input  $x$  exceeds the threshold value  $x_{th}$ , <sup>otherwise it is</sup> but  $g1$  [if not]. Therefore, the embodiment of Fig. 8(b) ✓  
is suitable when the memory cell used is binary. Fig. 8(c) shows an embodiment having tetral outputs  $ga$  and  $gb$  between the outputs  $g1$  and  $g2$ . The present embodiment is a suitable example when the tetral memory cells are used. The gap between the elements  $g1$ ,  $ga$ ,  $gb$  and  $g2$  can naturally be changed, if necessary, although they are shown equally distant in Fig. 8(c).  
In <sup>the</sup> case <sup>where</sup> the memory cells are exemplified by those ✓  
capable of storing data having continuous values, i.e., the so-called "analog values", the nonlinear transfer function circuit D to be used may have the

characteristics shown in Fig. 2(d).

Fig. 8(d) shows one embodiment of the correspondences in the embodiment of Fig. 8(a) between the word line selecting method for determining the neuron output values of the final layer from the neuron output values of the input layer and the write destination addresses. The operations of Fig. 8(a) will be described in the following with reference to Fig. 8(d). The neuron output values  $V_{11}$ ,  $V_{21}$ , - - -, and  $V_{n1}$  of the input layer are written in advance in the input/output circuit (although omitted from Fig. 8) in the memory cells on the word line of  $S = 1$  of the array A. First of all, the word lines of  $s = 1$  of the array A and the word lines of  $(s, j) = (1, 1)$  of the array B are selected simultaneously, although not necessarily completed. Then, the neuron output values  $V_{11}$ ,  $V_{21}$ , - - -, and  $V_{n1}$  of the input layer are outputted to the data lines of  $i = 1, 2, - - -,$  and  $n$  of the array A. On the other hand, the connection weights  $T^{111}$ ,  $T^{112}$ , - - -, and  $T^{11n}$  are outputted to the data lines of  $i = 1, 2, - - -,$  and  $n$  of the array B. These values are inputted to the multipliers  $m1, m2, - - -,$  and  $mn$  so that their products  $T^{111}V_{11}$ ,  $T^{112}V_{21}$ , - - -, and  $T^{11n}V_{n1}$  are inputted to the adders  $c1, c2, - - -,$  and  $cn$ . The results  $(T^{111}V_{11} + T^{112}V_{21}, - - -, + T^{11n}V_{n1})$  are inputted to the nonlinear transfer function circuit D. The output of this nonlinear transfer function circuit D is written through the write circuit WR (although omitted) in the memory cells corresponding to the write destination addresses of  $(s, i) = (2, 1)$  in the array A. Thus, the value of the first neuron output value  $V_{21}$  of the second layer is calculated. Next, the word line of  $s = 1$  of the array A and the word line of  $(s, j) = (1, 2)$  of the array B are simultaneously selected. Then, the neuron output values

$V_{11}$ ,  $V_{21}$ , - - -, and  $V_{n1}$  of the input layer are output-  
 putted to the data lines of  $i = 1, 2, - - -,$  and  $n$  of  
 (the) array A. On the other hand, the connection  
 weights  $T^1_{21}$ ,  $T^1_{22}$ , - - -, and  $T^1_{2n}$  are outputted to  
 5 the data lines of  $i = 1, 2, - - -,$  and  $n$  of (the) array  
 B. These values are inputted to the multipliers  $m_1$ ,  
 $m_2$ , - - -, and  $m_n$  so that their products  $T^1_{21}V_{11}$ ,  
 $T^1_{22}V_{21}$ , - - -, and  $T^1_{2n}V_{n1}$  are inputted to the adders  
 $c_1$ ,  $c_2$ , - - -, and  $c_n$ . The results  $(T^1_{21}V_{11} +$   
 10  $T^1_{22}V_{21}$ , - - -,  $+ T^1_{2n}V_{n1})$  are inputted to the non-  
 linear transfer function circuit D. The output of  
 this nonlinear transfer function circuit is written  
 through the write circuit (although not shown) in the  
 memory cells corresponding to the write destination  
 15 addresses of  $(s, i) = (2, 2)$  in (the) array A. Thus,  
 the value of the second neuron output value  $V_{22}$  of the  
 second layer is calculated. All the neuron output  
 values can be calculated by continuing the operations  
 thus far described according to Fig. 8(d). According  
 20 to the present embodiment, one neuron output value can  
 be determined by executing the read[ing] and writ[ing]  
 operations once in the arithmetic mode so that the  
 neural network processing can be accomplished at [a]  
 high speed. [Since] <sup>M</sup> moreover, <sup>since</sup> the arithmetic circuit  
 25 can be shared among all the neurons, [a] high integra-  
 tion can be <sup>attained</sup> made. Incidentally, Fig. 8(d) shows <sup>only</sup> one  
 example of <sup>the</sup> assignment of the memory cells, and the  
 present invention should not be limited thereto but  
 [can] <sup>may</sup> be modified in various manners. For example, the  
 30 plural <sup>xy of</sup> input data can <sup>may</sup> be continuously process[ed], as  
 has been described [above] hereinbefore. In this case, a  
 plurality of sets of neuron output values of an input  
 layer are required. For this operation, the neuron  
 output values of the input layer corresponding to the  
 35 plural <sup>xy of</sup> input data may be written in advance on the

plurality of different word lines of the array A so that they may be <sup>used</sup> consecutively [used]. Thus, the neuron output values of the input layer need not be read in for each data processing so that the data processings can be continuously accomplished at the high speed.

5 [Here is <sup>In this example,</sup> used one memory cell <sup>is used</sup> for storing the neuron output value and the connection weight. This allows only binary values to be taken as the neuron output values and the connection weights in <sup>the</sup> case <sup>where</sup> the binary memory cells are used. By using the multi-valued memory cells, as <sup>was</sup> <sup>above</sup> [has been] described <sup>herein-</sup> before, the neuron output values and the connection weight values could be increased, but the multi-valued

memory cells may have <sup>their</sup> [its] reliability <sup>decreased</sup> [made deficient] <sup>due to</sup> by the <sup>with the</sup> problem of S/N ratio. In this case, a plurality of memory cells <sup>may</sup> [can] be used for storing the neuron output values and the connection weights, as will be described <sup>below</sup> in the following.

Fig. 9(a) shows one embodiment of the case in which a <sup>number</sup>  $p$  [number] of memory cells are used for storing one neuron output value and in which a <sup>number</sup>  $q$  [number] of memory cells are used for storing one connection weight. The suffix  $i$ ,  $j$  or  $s$  appearing in Fig. 9(a) to indicate the neuron output value or the connection weight corresponds to that of the embodiment shown in Fig. 8. In the embodiment of Fig. 9(a), the <sup>number</sup>  $p$  [number] of continuous memory cells on one word line in the array A express <sup>as</sup> one neuron output value, and the <sup>number</sup>  $q$  [number] of continuous memory cells on one word line in the array B express <sup>as</sup> one connection weight.

The calculations of the neuron output values are carried out in the following manner. First [of all], like the embodiment of Fig. 8, the word line of  $s = 1$  of the array A and the word line of  $(s, j) = (1, 1)$  of the array B are <sup>selected</sup> simultaneously [selected]. Then, to the

data line group of  $i = 1, 2, \dots$ , and  $n$  composed of  $p$  number data lines of [the] array A, there [are] out-  
 putted the data expressing the neuron output values of  $V_{11}, V_{21}, \dots$ , and  $V_{n1}$  of the input layer, which are  
 5 inputted group by group to the adders  $a_1, a_2, \dots$ ,  
 On the other hand, the data expressing the connection weights  $T_{11}^1, T_{12}^1, \dots, T_{1n}^1$ , which are read in group  
 and an. To the data line group of  $i = 1, 2, \dots$ , and  $n$  composed of  $q$  number of data lines of the array  
 by group to the adders  $b_1, b_2, \dots$ , and  $b_n$  are output to the data line group of  
 $i = 1, 2, \dots$ , and  $n$  which is composed of  $q$  number of data lines  
 B, on the other hand, there are outputted the data ex-  
 pressing the connection weights  $T_{11}^1, T_{12}^1, \dots$ , and  
 10  $T_{1n}^1$ , which are inputted group by group to the adders  
 $b_1, b_2, \dots$ , and  $b_n$ . By the [aforementioned] adders  
 $a_1, a_2, \dots$ , and  $a_n$ , and  $b_1, b_2, \dots$ , and  $b_n$ , [the]  
 neuron output values  $V_{11}, V_{21}, \dots$ , and  $V_{n1}$  and the  
 connection weights  $T_{11}^1, T_{12}^1, \dots$ , and  $T_{1n}^1$  are  
 15 by the adders  $a_1, a_2, \dots, a_n$ , and  $b_1, b_2, \dots, b_n$  mentioned above  
 composed and inputted, as shown, to the multipliers  
 $m_1, m_2, \dots$ , and  $m_n$  to produce the products  $T_{11}^1 V_{11},$   
 $T_{12}^1 V_{21}, \dots$ , and  $T_{1n}^1 V_{n1}$ . These products are in-  
 putted to the adders  $c_1, c_2, \dots$ , and  $c_n$  so that  
 their results  $(T_{11}^1 V_{11} + T_{12}^1 V_{21}, \dots, + T_{1n}^1 V_{n1})$   
 20 are inputted to the nonlinear transfer function cir-  
 cuit D. The output of the nonlinear transfer function  
 circuit is written through the write circuit WR (al-  
 though not shown) to the  $p$  number of memory cell  
 groups corresponding to the write destination address  
 25  $(s, i) = (2, 1)$  in [the] array A. The output values of  
 all the neurons can be determined by continuing simi-  
 lar operations by using the same address as that of  
 Fig. 8(d).

Since, according to the aforementioned embodi-  
 30 ment, one neuron output value is expressed with [the]  $p$   
 number of continuous memory cells on one word line in  
 [the] array A, the multi-valued neuron output values can  
 be expressed by using the binary memory cells. Since,  
 moreover, one connection weight is expressed with [the]  
 35  $q$  number of continuous memory cells on one word line



in the array B, the multi-valued connection weights  
can be expressed by using the binary memory cells. As  
a result, the multi-valued<sup>multiple values</sup> such as the neuron output  
values or the connection weights, can be expressed<sup>using</sup> with  
5 the binary memory cells. In the aforementioned em-  
bodiment, moreover, the frequency of switching the ad-  
dresses is identical to that of the embodiment of Fig.  
8 so that the data can be processed at a high<sup>er</sup> speed  
like<sup>n</sup> the embodiment of Fig. 8. In order to write the  
10 result of the nonlinear transfer function circuit in  
the p number of memory cells expressing the neuron  
output values, the p number of writing operations may  
be continuously executed but can be easily accomplish-  
ed in parallel by providing the p number of write cir-  
15 cuits. In this case, it is possible to avoid the time  
losses for writing<sup>to</sup> the plural<sup>ty of</sup> memory cells. By pro-  
viding a plurality of read circuits, on the other  
hand, it is naturally possible to increase the speed  
of the reading operations. Incidentally, if a multi<sup>multiple</sup>  
20 input circuit is used as<sup>for</sup> the multipliers m1, - - -,  
and mn in the embodiment of Fig. 9(a), a similar cir-  
cuit can be realized without providing the adders a1,  
- - -, and an, and b1, - - -, and bn. In addition,  
the structure of the arithmetic circuit can be modi-  
25 fied in various manners.

The embodiment of Fig. 9(a) uses the p number of  
memory cells for storing the neuron output values and  
the q number of memory cells for storing the connec-  
tion weights. In other words, the neuron output  
30 values are expressed in the p bits, and the connection  
weights are expressed in the q bits. Since there are  
various methods of expressing the data in<sup>a</sup> the plural<sup>ity of</sup>  
bits, an expression method may be selected, if neces-  
sary, and it is accordingly sufficient to design the  
35 characteristics of the adders, the multipliers and the

nonlinear transfer function circuits. For example, the neuron output values can be expressed with the number of [such ones of the] memory cells of  $p$  bits expressing the neuron output values [as] <sup>high</sup> have a content of 1. Fig. 9(b) shows one embodiment of the input/output characteristics of the nonlinear transfer function circuit D suited for such <sup>case</sup>. In Fig. 9(b), characters  $g_1, g_2, \dots, g_p$  designate the  $p$  number of outputs of the nonlinear transfer function circuit D. Their individual outputs take the value 0 or 1, which is written in the  $p$  number of corresponding memory cells through the write circuit WR (although not shown). The outputs  $g_1, g_2, \dots, g_p$  take the value 1, when the individual inputs exceed the values  $x_{th1}, x_{th2}, \dots, x_{thp}$ , and otherwise take the value 0. These values  $x_{th1}, x_{th2}, \dots, x_{thp}$  may be at an equal or arbitrary distance between the upper limit  $x_1$  and the lower limit  $x_2$  of the inputs. If the distance between the values  $x_{thk}$  and  $x_{thk+1}$  ( $k = 1, \dots, p-1$ ) is set, as shown in Fig. 9(c), the nonlinear transfer function circuit  $g$  can be <sup>shown</sup> realized to have [the] sigmoid characteristics. According to the present embodiment, one neuron output value can be given [the]  $p$  number of values by the  $p$  number of memory cells. In the present embodiment, the  $p$  number of memory cells expressing the neuron output values are equivalently handled. <sup>it does not matter for</sup> Specifically, the influences upon the neuron output values [are equal no matter what of] <sup>to far</sup> the  $p$  number of memory cells [might] have its data inverted or fixed. As a result, the influences upon the neuron output values due to the fall of the data of one memory cell can be made lower than those of the general binary expressions. This expression method will be referred to as the "equivalent expression" in the following. [Here has been described] <sup>to far</sup> the neuron

output values<sup>have been described</sup>, but the aforementioned equivalent expression can naturally be used for the connection weights. ✓

The binary expressions can naturally be used.

5 This case is suited for expressing many values with less memory cells because the  $2^p$  values can be expressed in ~~the~~ p bits. Fig. 10(a) shows one embodiment of ✓  
the case, in which the binary expressions are used for ✓  
the neuron output values and ~~the~~ connection weights. ✓  
10 Only the memory cells on the data line of  $i = h$  ( $h = 1, 2, - - -,$  and  $n$ ) are shown in ~~the~~ array A, and only ✓  
the memory cells on the data line of  $i = h$  and on the word line of  $s = f$  ( $f = 1, 2, - - -,$  and  $m-1$ ) are ✓  
shown in ~~the~~ array B. In Fig. 10(a), letters WT des- ✓  
15 ignate a weighing circuit for weighing the signals of the memory cells and transmit<sup>ing</sup> them to the adders,  $ah$  ✓  
and  $bh$ . Here, the weighing coefficient<sup>s</sup> are varied for ~~the~~ every memory cell<sup>s</sup>, as shown, from 1 to  $2^p$  for the ✓  
neuron output values and from 1 to  $2^q$  for the connection weights. As a result, the neuron output values ✓  
20 and the connection weights to be input~~ted~~ to the multiplier  $mh$  can take ~~the~~  $2^p$  and  $2^q$  numbers of values, ✓  
respectively. The address selecting method for calculating the neuron output values may follow Fig. 8(d) ✓  
25 like the embodiment of Fig. 9. Fig. 10(b) shows one embodiment of the characteristics of the nonlinear transfer function circuit D in the embodiment of Fig. 10(a). The output  $g1$  alternately repeats the values 0 and 1, each time the input varies by  $(x_2 - x_1)/2^p$ ,  
30 and the output  $g2$  alternately repeats the values 0 and 1 for a period twice as long as that of the output  $g1$ . The periods are likewise varied by times so that the output  $g_p$  is set from 0 to 1 across the value of  $(x_2 - x_1)/2$ . In other words, the nonlinear transfer function circuit D may be so designed that it may operate  
35

as an A/D converter. In the present embodiment, too, the nonlinear transfer function circuit D can be so designed that the neuron output values may increase nonlinearly for the input. In order to increase the neuron output values according to the sigmoid function in response to the input, for example, the periods for the individual outputs ~~to vary~~ may be decreased with the increase in the input, ~~while~~<sup>with</sup> the period ratios between the different values ~~of~~ being held constant, so that the periods may be increased with the increase in the input when the input exceeds the value of  $(x_2 - x_1)/2$ . As has been described hereinbefore, according to the embodiments shown in Figs. 10(a) and 10(b), the neuron output values and the connection weights can be expressed to have the  $2^p$  values and the  $2^q$  values, respectively, by using the p and q numbers of memory cells, respectively. Thus, the embodiments are suited for giving the neuron output values and the connection weights multiple values with less memory cells. Incidentally, in the present embodiment, too, various modifications can naturally be made such that the functions of the weight circuit WT and the adders  $a_1$ , - - -, and  $a_n$ , and  $b_1$ , - - -, and  $b_n$  are given to the multipliers by using multi-input circuits ~~as~~<sup>with</sup> the multipliers  $m_1$ , - - -, and  $m_n$ . ~~Here has been described~~<sup>have been described</sup> the embodiments using the equivalent expressions and the binary notations. In addition, there are various methods such as the method of expressing a negative number with a code bit or a method of expressing data in <sup>a</sup>plural <sup>ky of</sup> bits, which can be separately used, if necessary.

Next, ~~here~~ will be described an embodiment, in which a dynamic memory cell (i.e., DRAM cell) composed of one MOS transistor and one capacitor is used in the memory.

Fig. 11 shows an embodiment in which the embodiment shown in Fig. 8(a) is composed of the DRAM cells. In Fig. 11, ~~the~~ array A and ~~the~~ array B are <sup>comprised</sup> composed of: a plurality of intersecting data line pairs DA1 and  $\overline{DA1}$ , - - -, and DAN and  $\overline{DAN}$ , and DB1 and  $\overline{DB1}$ , - - -, and DBn and  $\overline{DBn}$ , and word lines WA1, WA2, - - -, and WAm, and WB1,1, WB1,2, - - - WB1,n, and WB2,1, WB2,2, - - -, and WBm-1,n; and memory cells MC disposed at their intersections. Each memory cell MC is arranged at ~~(either of)~~ the intersections between the paired data lines and the word line. Thus, the embodiment has the so-called "folded-bit line structure". In Fig. 11, letters PR, SA, RSA and WS designate a precharge circuit, a sense amplifier, a read sense amplifier, and a write switch, respectively, and correspond to the array control circuits 13A and 13B of Fig. 7(b). Letters MT appearing in the arithmetic circuit designate a multiplier. Numeral 16 designates a clock generator for generating clocks  $\Phi_A$  and  $\Phi_B$  for controlling other circuits in response to addresses ADDA and ADDB and chip select signals  $\overline{CSA}$  and  $\overline{CSB}$  <sup>obtained externally</sup> given from the outside of the chip. ✓

In the following, the operations of the embodiment of Fig. 11 will be described with reference to Fig. 12 and Figs. 13(a) and 13(b). Fig. 12 shows one embodiment between the operation modes and the external signals. As has been described hereinbefore, the word lines of ~~the~~ memory cell arrays A and B are selected one by one in the earlier half of the arithmetic mode, and one word line of ~~the~~ memory cell array A is selected in the later half. In the memory mode, on the other hand, ~~the~~ memory cell arrays A and B are independently subjected to the reading and writing operations. In Fig. 12, the operation modes are further divided so that those controls may be facilitat-

ed. Letters AR, AW, BR and BW in the memory mode <sup>distinguish</sup> designate the reading mode <sup>from</sup> the array A <sup>from</sup> the writing mode in the array, <sup>from</sup> the reading mode <sup>from</sup> the array B <sup>from</sup> and the writing mode in the array B, respectively. On

5 the other hand, letters NR and NW in the arithmetic mode <sup>designate</sup> the earlier half for reading and calculating the individual data and the later half for writing the arithmetic results, respectively. In order to switch those six modes, the present embodiment uses four external input signals: chip select signals  $\overline{CSA}$  and  $\overline{CSB}$ ;

10 write control signal  $\overline{WE}$ ; and arithmetic circuit control signal  $\overline{NE}$ . The chip select signals  $\overline{CSA}$  and  $\overline{CSB}$  assign the selections of the chips and the arrays A and B. The chips are unselected if both the signals

15  $\overline{CSA}$  and  $\overline{CSB}$  are at the H (i.e., high) level; the array is selected if the signal  $\overline{CSA}$  is at the L (i.e., low) level; and the array B is selected if the signal  $\overline{CSB}$  is at the L (i.e., low) level. The write control signal  $\overline{WE}$  is <sup>used</sup> one for switching the write and read, as has

20 been described hereinbefore, and establishes the reading operation at the H level and the writing operation at the L level. The signal  $\overline{NE}$  also establishes the memory mode at the H level and the arithmetic mode at the L level, as has been described hereinbefore. If,

25 <sup>for example</sup> [therefore], both the signals  $\overline{CSA}$  and  $\overline{CSB}$  are at the L level and the signal  $\overline{WE}$  is at the H level whereas the signal  $\overline{NE}$  is at the L level, <sup>is established</sup> [for example] there is established the earlier half mode NR of the arithmetic mode, in which both the arrays A and B are read out.

30 Since the switching of the arrays A and B is assigned by the chip select signal, the address signal can divide the addresses into the address group ADDA for selecting the memory cells of the array A and the address group ADDB for selecting the memory cells of

35 the array B. Here, the address group ADDA is the

generic name of the X-addresses for selecting the word lines of the array A and the Y-addresses for selecting the data lines of the array A. Likewise, the address group ADDB is the generic name of the X-addresses for selecting the word lines of the array B and the Y-addresses for selecting the data lines of the array B. In each operation mode, these address groups are applied to the address pins in accordance with Fig. 12. According to the embodiment of Fig. 12 thus far described, the two chip select signals are provided to switch ~~the~~ arrays A and B thereby ~~to~~ separate<sup>ing</sup> the addresses between ~~the~~ arrays A and B. Since ~~the~~ arrays A and B can be independently selected, it is possible to control each operation mode easily ~~for~~<sup>ing</sup> selecting ~~the~~ arrays A and ~~^~~ B. Incidentally, the relations~~e~~ between the operation modes~~e~~ and the external signals~~s~~ can naturally be modified in various manners in addition to those of Fig. 12. For example, ~~there can be~~<sup>one can</sup> adopt~~e~~: the method of adding addresses for switching the arrays A and B by using the chip select signal CS only; or the method of generating the X-address for selecting the word line of ~~the~~ array B in the mode NR from the counter disposed in the chip, by not dividing the addresses for ~~the~~ arrays A and B but selecting either ~~the~~ array A or B only.

Fig. 13(a) shows an embodiment of the waveforms of Fig. 11 in the memory mode, and Fig. 13(b) shows an embodiment of the waveforms of Fig. 11 in the arithmetic mode.

The operations of the memory mode are similar to the read~~ing~~ and writ~~ing~~<sup>e</sup> operations of the ordinary DRAM. Fig. 13(a) shows the voltage waveforms in<sup>the</sup> case<sup>where</sup> the read~~ing~~ operations (in the mode AR) and the writ~~ing~~<sup>e</sup> operations (in the mode AW) are continuously executed for the memory cell at the intersection between

the word line  $W_{A1}$  and the data line  $DA_{11}$  in the array A in the memory mode. In Fig. 13(a), letters  $V_{cc}$  designate a positive supply potential. Since, in the memory mode, the arithmetic circuit control signal  $\overline{NE}$  is at the high level, the arithmetic circuit start signal  $\Phi_N$  is fixed at the low level so that the arithmetic circuit 12 is OFF. Before the start of the reading operation, signals  $PPA$  and  $PNA$  are set <sup>to</sup>  $V_{cc}/2$  so that the sense amplifier SA is OFF. Since a precharge signal  $\Phi_{PA}$  is at the high potential, on the other hand, the precharge circuit PR is turned on to short the data line pairs  $DA_{11}$  and  $\overline{DA_{11}}$ , - - -, and  $DA_n$  and  $\overline{DA_n}$  and to set the potential at a precharge potential  $V_H$ . This precharge potential  $V_H$  is set <sup>to</sup>  $V_{cc}/2$  according to the so-called "half-precharge method". When the chip select signal  $\overline{CSA}$  drops to the low potential, the precharge signal  $\Phi_{PA}$  falls to turn off the precharge circuit PR so that the word line  $W_{A1}$  selected by the address signal  $ADDA$  and a read Y-Select signal  $Y_{RA1}$  are transited to the high potential. As a result, the MOS transistors of all the memory cells MC connected with the word line  $W_{A1}$  are rendered conductive <sup>individually</sup> to establish individually delicate potential differences on the data line pairs  $DA_{11}$  and  $\overline{DA_{11}}$ , - - -, and  $DA_n$  and  $\overline{DA_n}$  in accordance with the electric charges stored in the capacitor. This potential difference is read out and is detected by the sense amplifier RSA'ed with the Y-Select signal  $Y_{RA1}$  so that it is converted into the impedance difference of read lines  $OA$  and  $\overline{OA}$ . This impedance difference is converted by the input/output circuit into a voltage difference, which is amplified so that the content of the memory cell, i.e., the potential corresponding to 1 or 0 is outputted as the read data  $DO$ . The so-called "rewriting operation" is executed in parallel



with the aforementioned operations in the following manner. After the individual delicate potential difference<sup>a</sup> have been established in the data line pairs  $\overline{DA1}$  and  $\overline{DA1}$ , - - -, and  $\overline{DAn}$  and  $\overline{DAn}$ , the signal PPA is transited to the high potential whereas the signal PNA is transited to the low potential to start the sense amplifier SA. As a result, the delicate potential difference established in the data line pairs is amplified to transit the data lines at the high potential to the level  $V_{cc}$  and the data lines at the low potential to 0 V. As a result, the capacitors of all the memory cells MC connected with the word line  $\overline{WA1}$  are written again with the potential corresponding to the data before read. When the chip select signal  $\overline{CSA}$  takes the high potential after the end of the rewriting operation, the selected word line  $\overline{WA1}$  and the read Y-select signal  $\overline{YRA1}$  are transited to the low potential, and the signals PPA and PNA are then transited to  $V_{cc}/2$  to turn off the sense amplifier SA and transit the precharge signal  $\Phi_{PA}$  to the high potential. As a result, the data line pairs are shorted, and the precharge potential  $V_H$  is set with the potential, until the initial state is restored. The operations thus far described are the reading operations.

Subsequently, the operations are shifted to the writing operations (in the mode AW) of the same cells. When, in the writing operations, the chip select signal  $\overline{CSA}$  takes the low potential and the write control input  $\overline{WE}$  takes the low potential, the data given to the write data  $\overline{DI}$  are written in the selected memory cell in the array A. In the writing operation, too, the precharge signal  $\Phi_{PA}$  is dropped at first to turn off the precharge circuit PR when the chip select signal  $\overline{CSA}$  drops to the low potential. Next, the word line  $\overline{WA1}$  selected by the address signal  $\overline{ADDA}$  and the

Y-Select signal YRA1 are transited to the high potential. As a result, the MOS transistors of all the memory cells MC connected with the word line WA1 are rendered conductive so that the delicate potential differences are <sup>individually</sup> established in the data line pairs DA1 and  $\overline{DA1}$ , - - -, and DAN and  $\overline{DAN}$  in accordance with the electric charges stored in the capacitor. The delicate potential differences established in the data line pairs are amplified by the sense amplifier SA. Subsequently, an input circuit starting signal  $\Phi_{WRA}$  generated as a result of <sup>the</sup> transition of the control input  $\overline{WE}$  to the low potential is transited to the high potential. As a result, the data given to the write data DI are transmitted as the difference <sup>between</sup> signals to write line pairs IA and  $\overline{IA}$ . Moreover, the write Y-select signal YWA1 is transited to the high potential to turn on the write switch WS connected with the write destination memory cell. As a result, the write line pairs IA and  $\overline{IA}$  are conducted to the data line pairs DA1 and  $\overline{DA1}$ , respectively. As a result, the data line pairs DA1 and  $\overline{DA1}$  are set to the potential corresponding to the data fed to the write data DI. After this, the input circuit starting signal  $\Phi_{WRA}$  is transited to the low potential, but the potential of the data line pairs is held by the sense amplifier SA. In the data line pairs for which the write switch WS is not turned on, the signal read out at first is amplified as it is by the sense amplifier so that the rewrite is executed. When the chip select signal  $\overline{CSA}$  takes the high potential after the end of the rewriting operation, the selected word line WA1 and the write Y-selection signal YWA1 are shifted to the low potential. After this, the signals PPA and PNA are transited to  $V_{CC}/2$  to turn off the sense amplifier SA and to transit the precharge signal  $\Phi_{PA}$

to the high potential. As a result, the data line pairs are shorted and <sup>have their</sup> [has its] potential set to the precharge potential  $V_H$  so that the initial state is restored. The operations [thus far] <sup>above</sup> described are the writing operations.

Here, the foregoing description is directed to the case in which a memory cell in the array A is continuously subjected to the reading operation and the writing operation. Despite [of] this description, however, one of the reading and writing operations can be continuously executed. Moreover, it is quite natural that memory cells in desired positions in a memory cell array, which are different for every reading operation or writing operation, can be subjected to the reading operation or the writing operation by switching the modes AR, AW, BR and BW.

Next, the operations of the arithmetic mode will be described [in the following]. Fig. 13(b) shows the operation waveforms for obtaining a neuron output value  $V_{12}$ . [Let it be] <sup>assumed</sup> that the necessary connection weights and neuron output values or the like have already been written by the writing operations in the memory mode. In order to establish the mode NR, first [of all], the chip select signals  $\overline{CSA}$  and  $\overline{CSB}$  are set to the low level, and the write control signal  $WE$  is set to the high level whereas the arithmetic circuit control signal  $\overline{NE}$  is set to the low level. The addresses ADDA and 'ADDB are so set as to select the word line WA1 of the array A and the word line WB1 of the array B. Since the signals  $\overline{CSA}$  and  $\overline{CSB}$  are at the low level, the precharge signals  $\Phi_{PA}$  and  $\Phi_{PB}$  are transited to the low level. Since the signal  $\overline{NE}$  is at the low level, the arithmetic circuit starting signal  $\Phi_N$  is transited to the high level. Subsequently, the word lines WA1 and WB1 are selected so that the neuron

output values  $V_{11}$ ,  $V_{21}$ , - - -, and  $V_{n1}$  and the connection weights  $T^{11}_1$ ,  $T^{12}_1$ , - - -, and  $T^{1n}_1$  are read out of the memory cells <sup>through</sup>  $(on)$  the word line  $W_{A1}$  onto the data lines. Thus, the neuron output values read out from

5 the array A and the connection weights read out from the array B are inputted to the multiplier MT which has been started by the arithmetic circuit starting signal  $\Phi_N$ , as shown in Fig. 11. In the multiplier

10 MT, the data lines at the side of  $(the)$  array A and the data lines at the side of  $(the)$  array B are individually connected with the gates of the MOS transistors ( $T_1$ ,  $T_2$ ), which in turn are connected with the product/sum

15 output line NO and the dummy line DM through the switching MOS transistor ( $T_3$ ) fed with the arithmetic circuit starting signal  $\Phi_N$ . The product/sum output line NO has its one terminal connected with the power source VM through the load  $RM_1$ , and the dummy line DM

20 has its one terminal <sup>grounded to</sup> earthed to the ground. When the signal read out to the data lines is amplified to  $V_{cc}$  or 0 V by the sense amplifier SA, the electric current  $(flows)$   $(through the load RM_1)$  from the power source <sup>through the load RM\_1, and</sup> VM to the earthed electrode in the multiplier, in which the product  $(to)$  the neuron output value and the connection

25 weight is 1. As a result, the potential of the product/sum output line NO drops in proportion to the number of combinations, in which the product of the neuron output value and the connection weight is 1. The product/sum NO is inputted to the nonlinear transfer function circuit D. In this nonlinear transfer

30 function circuit D, the summation of the products of the neuron output values and the connection weights is so high that the detection result of whether or not the potential of the product/sum output line NO is lower than reference voltage VR is outputted to <sup>the</sup> a line

35 NV. In the waveforms of the product/sum output line

NO shown in Fig. 13(b), solid lines show the case in which the result of the product sum is small, and broken curves show the case in which the result of the product sum is large. The input/output circuit detects the result of the nonlinear transfer function circuit D and outputs the neuron output value  $V_{12}$ , which is to be subsequently written in the memory cells, to the write line pairs IA and  $\overline{IA}$ . Fig. 13(b) show the waveforms of the write line pair IA. This pair IA takes the high level, as indicated by a broken curve, in case the product sum is large, and the low level, as indicated by a solid curve. In case the product sum is small, the word line pair  $\overline{IA}$  takes an opposite phase. At the time when the neuron output values are outputted to the write line pairs IA and  $\overline{IA}$ , a latch signal  $\Phi_L$  is transited to the high potential. As a result, the potentials outputted to the write line pairs IA and  $\overline{IA}$  are latched by a latch circuit which is disposed in the input/output circuit IO. The latch signal  $\Phi_L$  may be raised with a delay till the signals appear in the pairs IA and  $\overline{IA}$ , in response to the fall of the arithmetic circuit starting signal  $\overline{NE}$ . Subsequently, the arithmetic circuit starting signal  $\Phi_N$  is transited to the low potential to turn off the arithmetic circuit so that the data lines are precharged as in the memory mode after the word lines have fallen. At this time, the latch signal  $\Phi_L$  is left at the high potential so that the neuron output values outputted to the write line pairs IA and  $\overline{IA}$  are held constant.

Next, the mode shifts to the  $NW$  mode  $[NW]$ , i.e., the later half of the arithmetic mode. First of all, the chip select signal  $\overline{CSA}$  and the write control signal  $\overline{WE}$  are set to the low level, and the chip select signal  $\overline{CSB}$  is set to the high level in order to switch the address

ADDA so that the memory cell for writing the neuron output value in the array A may be selected. The arithmetic circuit starting signal  $\overline{NE}$  is left at the low level. As a result of the fall of the signal  $\overline{CSA}$ , the precharge signal  $\Phi_{PA}$  drops to the low level to establish a state <sup>in which</sup> for the array A <sup>may</sup> to be written. Subsequently, the potentials of the selected word line WA2 and the write Y-selection signal YWA1 fall. As a result, the neuron output value  $V_{12}$ , outputted to the write line pairs IA and  $\overline{IA}$ , is written in the memory cell which is connected with the WA2 and the data line DA1. Finally, all the potentials of the word lines are dropped for the precharge. Since, moreover, the arithmetic circuit control signal  $\overline{NE}$  falls, the latch signal  $\Phi_L$  falls to release the latch. Thus, preparations are made for the next operations. The operations thus far described <sup>thus far</sup> are <sup>all</sup> those in the arithmetic mode. All the neuron output values can be calculated by continuing similar operations in different addresses in accordance with Fig. 8(d).

In the structure thus far described, the circuit of the multiplier MT, which is connected with the dummy line DM, may be omitted. If, however, the gate capacitors, or the like, of the MOS transistors of the multiplier MT are added to the data lines only at one side, the data line capacities are unbalanced (to) <sup>which</sup> trouble the operations of the sense amplifier, <sup>whatever</sup> as the case may be. <sup>Using this scenario,</sup> [In this case, the structure of Fig. 11 could avoid] the inferior influences coming from the unbalance of the data line capacities. <sup>could be avoided in Fig 11.</sup>

Next, an embodiment of the circuit suitable for use in Fig. 11 is shown. Fig. 14(a) shows one embodiment of the nonlinear transfer function circuit D. The present embodiment is constructed of: a differential amplifier composed of bipolar transistors Q720

and Q719, a resistor R72, and a MOS transistor Q721;  
and an inverter composed of an inverter INV75, MOS  
transistors Q715, Q716, Q717 and Q718, a resistor R71  
and a diode D71. The present circuit is<sup>initially</sup> started when  
5 the signal  $\Phi_N$  takes the high potential. Fig. 14(b)  
shows a relation between the potential of the product/  
sum output line NO,<sup>[or]</sup> the input of the nonlinear  
transfer function circuit D, and the potential of the  
output NV. The output NV takes the high potential, if  
10 the potential of the product/sum output line NO is  
lower than the reference potential VR, but takes the  
low potential if the potential of the line NO is  
higher than the reference potential VR. Since, ac-  
cording to the present embodiment, <sup>[the]</sup> bipolar tran-  
15 sistors are used in the differential amplifier, it is  
possible to realize a nonlinear circuit which is  
characterized to have a steep rise for the change in  
the input. By setting the reference potential VR to a  
desired value, moreover, the characteristics of the  
20 nonlinear transfer function circuit D can be easily  
changed. Incidentally, the output of the different<sup>ial</sup>  
amplifier cannot be made so high so as to avoid the  
saturation of the bipolar transistor Q719. <sup>(As)</sup> a re-  
sult, the downstream inverter may not operate if it is  
25 connected directly with the output of the differential  
amplifier. Therefore, the resistor R71 and the diode  
D71 are provided to <sup>adjust</sup> drop the potential<sup>which is</sup> to be input<sup>ted</sup>  
to the MOS transistor Q717, <sup>lower</sup>.

Fig. 14(c) shows one embodiment of the input/out-  
30 put circuit IO. The write circuit WR is composed, as  
shown in Fig. 14(c), of an input buffer INBUF, write  
switches SWA and SWB, a latch circuit LAT, and in-  
verters INVIA and INVIB. The write switches SWA and  
SWB are used to switch<sup>whichever</sup> <sup>[which of the]</sup> arrays, A and<sup>or</sup> B, has  
35 its memory cells written with the write data DI. When

the switching signal  $\Phi_{WRA}$  is at the high potential, the write data DI is written through the input buffer INBUF <sup>to</sup> the memory cells of the array A by the write line pairs IA and  $\overline{IA}$ . When the switching signal  $\Phi_{WRB}$  is at the high potential, the write data DI is written through the input buffer INBUF <sup>to</sup> the memory cells of the array B by the write line pairs IB and  $\overline{IB}$ . The latch circuit LAT latches the data, which are outputted to the output NV of the nonlinear transfer function circuit D in the arithmetic mode, to write them in the memory cells of the array A by the write line pairs IA and  $\overline{IA}$ . Since the potential relations of the output NV of the nonlinear transfer function circuit D and the write line pairs IA and  $\overline{IA}$  are in phase, <sup>which</sup> as is apparent from <sup>Fig.</sup> 14(c), the potential relations <sup>hip</sup> between the product/sum output line NO of the nonlinear transfer function circuit D and the input common line IA <sup>is</sup> <sup>one another</sup> opposite to each other, as shown in Fig. 14(d). Since, in the embodiment of Fig. 11, the potential of the product/sum output line NO is the lower <sup>for</sup> the larger product sum of the neuron output values and the connection weights, as has been described hereinbefore, the circuit is made such that the potential relations <sup>hip</sup> between the product/sum output line NO and the input common line IA <sup>be</sup> opposite to each other. <sup>one another</sup> Since <sup>this</sup> in case the potential of the product/sum output line NO is <sup>so</sup> designed that it may <sup>increase</sup> rise the more for the larger product sum of the neuron output values and the connection weights, it is quite natural that the circuit may be <sup>so</sup> made that the potential relations <sup>hip</sup> between the product/sum output line NO and the input common line IA <sup>be</sup> in phase.

Fig. 14(e) shows one embodiment of the read circuit OUT. This read circuit OUT is composed of a current/voltage converter IVOUT1, a level shift cir-



cuit LS, a read latch circuit OUTLT and an output buffer BUFOUT. In the current/voltage converter IVOUT1, the data read out as the impedance differences to the read lines OA and  $\overline{OA}$  and the read lines OB and  $\overline{OB}$  are converted to the differences of the voltages of the lines OA' and  $\overline{OA}'$  <sup>and</sup> read lines OB' and  $\overline{OB}'$ . In the level shift LS, the voltage of the data read out from the current/voltage converter IVOUT1 is shifted to a level, at which the bipolar transistors in the downstream read latch circuit OUTLT are not saturated, to transmit it to the read latch circuit OUTLT.

A detailed embodiment of the read latch circuit OUTLT is shown in Fig. 14(f). The read differential amplifiers AMPA and AMPB in the read latch circuit OUTLT are used to switch which of the data read out from the memory cells of the array A through the read lines OA and  $\overline{OA}$  to lines L1 and L2 <sup>with</sup> and the data read out from the memory cells of the array B through the read lines OB and  $\overline{OB}$  to the lines L3 and L4 <sup>are</sup> are to be read out as the read data DO. When a switch signal  $\Phi_A$  is at the high potential, the data read out from the memory cells of the array A are outputted as the read data DO. When a switch signal  $\Phi_B$  is at the high level, the data read out from the memory cells of the array B are outputted as the read data DO. In the read latch circuit OUTLT, the bipolar transistor Q1A is turned off, but the bipolar transistor Q1B is turned on when the read latch circuit  $\Phi_{LR}$  is transited to a higher potential than the voltage  $V_{B2}$ . As a result, the differential amplifiers AMPA and AMPB are turned off, and the differential amplifier AMPC is turned on. As a result, the read data are latched by the differential amplifier AMPC and the level shift circuit LSC. Specifically, according to the present embodiment, the read data DO can be latched and con-

tinuously<sup>continuously</sup> output<sup>period of</sup>ted for a desired time [period] by trans-  
siting the read latch circuit  $\Phi_{LR}$  to a higher poten-  
tial than the voltage  $V_{B2}$  after the read data have  
been fixed.

5 Incidentally, in the embodiment of the multiplier  
MT shown in Fig. 11, the data line pairs of ~~the~~ array  
A are connected with the gates of the MOS transistors  
which are farther from the earthed electrode than the  
data line pairs of ~~the~~ array B. As a result,<sup>when a product is to be calculated,</sup> the neu-  
10 ron output values and the connection weights are not  
<sup>handled</sup> equivalently ~~handled~~ when a product is to be taken.

If this raises a problem, ~~an~~<sup>the</sup> embodiment ~~of~~<sup>in</sup> Fig. 15 may  
be used. In Fig. 15, the data line  $DA_i$  is connected  
with the gates of MOS transistors Q7C3 and Q7C6, and  
15 the <sup>paired</sup> data line ~~pair~~  $DB_i$  is connected with the gates of  
MOS transistors Q7C5 and Q7C4. Since the two data  
lines are connected with the MOS transistors closer to  
the earthed electrode and the MOS transistors farther  
from the same, the neuron output values and the con-  
20 nection weights are <sup>handled</sup> equivalently handled. As has been  
described hereinbefore, according to the embodiment  
shown in Fig. 11, the embodiment shown in Fig. 8(a)

can be realized by using the DRAM cell which is com-  
posed of one MOS transistor and one capacitor. The  
25 DRAM cell can have its occupied area drastically  
reduced<sup>in order</sup> to attain ~~a merit~~<sup>the benefit</sup> that it ~~can~~ be ~~realized in~~  
high<sup>ed</sup> integration<sup>although not discussed earlier,</sup> over the chip. In the DRAM cell com-  
posed of one transistor and one capacitor, ~~although~~  
not touched in the foregoing description, refreshing  
30 operations are necessary within a constant time period  
for compensating the reduction of the stored charges  
due to the current leakage of the capacitor. In the  
present invention, too, the refreshing operations can  
be easily accomplished, if necessary, like the ordi-  
35 nary DRAM no matter which ~~the~~ mode might be<sup>used</sup> the memory

mode or the arithmetic mode.

In the <sup>above mentioned</sup> aforementioned embodiment, the memory cells are exemplified by the DRAM cells but should not be limited thereto, but <sup>A</sup> a similar data processing system can also be realized by using even other memory cells. Next, an embodiment using SRAM cells will be described [in the following]. Figs. 16(a) and 16(b) are circuit diagrams showing SRAM cells MCS. In the embodiment of the present invention, the SRAM cells MCS are used as the MC of Figs. 7(a) and 7(b), Fig. 8(a), Fig. 9(a) and Fig. 10(a). Either of the embodiments, <sup>shown in</sup> of Figs. 16(a) and 16(b), is advantageous in that the controls are far easier than the case <sup>which uses</sup> of using the DRAM cells because it requires neither <sup>a</sup> rewriting nor refreshing operation [unlike the DRAM cells]. Fig. 16(c) shows one embodiment for realizing the embodiment of Fig. 8(a) by using the SRAM cell of Fig. 16(a) or 16(b). In Fig. 16(c), letters MCS designate the SRAM cell, and letters LD designate a data line load. Figs. 17(a) and 17(b) show examples of the operation waveforms. Fig. 17(a) shows an example of the case in which cells connected with the data lines DA1 and  $\overline{DA1}$  and the word line WA1 are continuously subjected to the reading operations and the writing operations in the memory mode. Fig. 17(b) shows an example of the operation waveforms of the case in which the neuron output valve  $V_{12}$  is to be calculated in the arithmetic mode from both the neuron output values  $V_{11}$ ,  $V_{21}$ , - - -, and  $V_{n1}$  <sup>which are</sup> stored in the memory cells of the word line WA1, and the connection weights  $T^{11}_1$ ,  $T^{12}_1$ , - - -, and  $T^{1n}_1$  <sup>which are</sup> stored in the memory cells of the word line WB1. The basic operations are similar to those of the aforementioned case of the DRAM cells, and their description will be omitted. The SRAM cell has <sup>the benefit</sup> a merit that its control is simpler than the DRAM

cell, because it does not need the rewriting operation and the refreshing operation. Because <sup>there is</sup> no necessity for the rewriting operation, moreover, <sup>benefit is</sup> there is another <sup>benefit is</sup> merit that the reading and writing speeds in the memory mode and the cycles in the arithmetic mode can be accelerated.

The description thus far made is directed to the example of the circuit structure, in which the embodiment of Fig. 8 is realized by using the DRAM cells and the SRAM cells. Next, <sup>will be described</sup> [here will be described] an example of the circuit structure for expressing the neuron output values and the connection weights by using a plurality of memory cells. Although the embodiment to be described uses <sup>will be described</sup> [the] DRAM cells, the present invention can <sup>be</sup> [be] likewise realized even <sup>be</sup> [by] using <sup>be</sup> [the] SRAM cells.

Next, <sup>will be described</sup> [here will be described] an example of the circuit structure in which the neuron output values are expressed by using the DRAM cells and in which the connection weights are expressed by using a plurality of memory cells. In Fig. 18(a), the data line pairs  $\overline{DA11}$  and  $\overline{DA11}$ ,  $\overline{DA12}$  and  $\overline{DA12}$ , - - -, and  $\overline{DA1P}$  and  $\overline{DA1P}$  in <sup>will be described</sup> [the] array A correspond to the data line pairs in <sup>will be described</sup> [the] array A, which are to be inputted to the adder  $a1$  in Fig. 9(a). On the other hand, the data line pairs  $\overline{DAn1}$  and  $\overline{DAn1}$ , and  $\overline{DAn2}$  and  $\overline{DAn2}$ , - - -, and  $\overline{DAnP}$  and  $\overline{DAnP}$  in <sup>will be described</sup> [the] array A correspond to the data line pairs in <sup>will be described</sup> [the] array A, which are to be inputted to the adder  $a2$  in Fig. 9(a). <sup>are provided</sup> [The] array B has similar correspondences. As shown in the input/output circuit DIO10, <sup>are provided</sup> [there are provided an] r number of input terminals  $\overline{DO1}$ , - - -, and  $\overline{DOr}$  and <sup>are provided</sup> [an] r number of output terminals  $\overline{DI1}$ , - - -, and  $\overline{DIr}$  (wherein r is the larger number of p and q) so that <sup>made up</sup> [the] data of p bits or q bits indicating the neuron output values or the connection weights

may be simultaneously read out or written in the memory mode. In the array A in the arithmetic mode, the data of every p bits read out to the data lines by selecting the word lines are synthesized by the adder ADD to output the neuron output values to the neuron output value output lines VO1, VO2, - - -, and VOn. In the array B, on the other hand, the data<sup>made up of</sup> of q bits read out to the data lines by selecting the word lines are synthesized by the adder ADD to output the connection weights to connection weight output lines TO1, TO2, - - -, and TOn. These values are inputted to the BLK2 so that the resultant product sum is inputted to the nonlinear transfer function circuit D10. The output of the nonlinear transfer function circuit D10 corresponding to the neuron output value is transmitted to the input/output circuit DIO10 and latched by the latch signal  $\Phi_L$ . Subsequently, the address is switches to select the p number of cells to write the determined neuron output value, and the write Y-select signal YWAi is raised to write the latched neuron output values in parallel in the p number of selected cells. By continuing these operations, the neuron output values can be updated like<sup>in</sup> the embodiment of Fig. 11(a). According to the present embodiment, the embodiment of Fig. 9(a) can be realized by equivalently adding the data of the plural<sup>pl.</sup> memory cells inputted to the adder ADD. By weighing and adding, bit by bit, the data of the plural<sup>pl.</sup> memory cells inputted to the adder ADD, moreover, it is possible to realize the embodiment of Fig. 10(a), in which the neuron output values and the connection weights are expressed with binary numbers of plural<sup>pl.</sup> bits. Since the present embodiment can also be applied to the case in which the neuron output values and the connection weights are expressed with a plurality of bits by another method,

a variety of data processings can be accomplished in accordance with the purpose. Since the DRAM cells are used in the present embodiment, a high integration can be achieved. Since, moreover, the data of the plural<sup>ity of</sup> memory cells are processed in parallel, both in the memory mode and in the arithmetic mode, [the] data processing can be executed at [a] high speed<sup>like</sup> the case<sup>in which</sup> [of] expressions<sup>have</sup> with 1 bit, although the neuron output values and the connection weights are expressed by the plural<sup>ity of</sup> bits. Here, in [the] BLK1, the signals of<sup>the</sup> plural memory cells are synthesized by the adder, and the result is input[ed] to [the] BLK2<sup>which</sup> act[ing] as the product/sum circuit. However, a variety of modifications can be made by omitting the addition at [the] BLK1<sup>such</sup> that<sup>io</sup> the data of the plural<sup>ity of</sup> memory cells indicating the neuron output values or the connection weights are input[ed] in parallel to the product/sum circuit BLK2 so that they may be subjected to [the] multiplications and summations.

In the following<sup>embodiment</sup>, Fig. 18(b) for realizing the embodiment of Fig. 9, in which the neuron output values and the connection weights are expressed in a plurality of equivalent bits by the embodiment shown in Fig. 18(a), shows one embodiment of [the] BLK1 of Fig. 18(a). [Here is shown] [the] BLK1<sup>as shown</sup> which is connected with [the] data lines DA11, - - -, and DA1P of [the] array A. The same circuit can also be used in another BLK1 of [the] array A. The circuit of the present embodiment can also be used in [the] array B, if the number of the data line pairs, the read line pairs or the write line pairs is changed from p to q and if a q number of circuits, each having a p number of precharge circuits PR, are provided. In the present embodiment, [there are provided] p pairs of read line pairs OA1 and OA1, - -, and OAp and OAp and p pairs of write line pairs IA1

and  $\overline{IA1}$ , - - -, and  $IAp$  and  $\overline{IAp}$ <sup>are provided</sup> so that the p number ✓  
of memory cells may be subjected to the writing or  
reading operations in parallel. The read sense ampli-  
fier RSA and the write switch WS are consecutively  
5 connected in the same BLK1, as shown, with the read  
line pairs  $OA1$  and  $\overline{OA1}$ , - - -, and  $OAp$  and  $\overline{OAp}$  and the  
p pairs of write line pairs  $IA1$  and  $\overline{IA1}$ , - - -, and  
 $IAp$  and  $\overline{IAp}$ . For one pair of read or write lines,  
specifically, ~~every~~<sup>each</sup> p pairs are<sup>is</sup> connected with the ✓  
10 data line pair. The adder ADD is composed of a load  
circuit LD103 and a p number of voltage/current con-  
verters VI. In the voltage/current converter VI, the  
data lines  $DA11$ ,  $DA12$ , - - -, and  $DA1p$  are connected  
with the gates of the MOS transistors, which in turn  
15 are connected in series with the MOS transistors hav-  
ing their gates fed with the arithmetic circuit start-  
ing signal  $\Phi_N$ , to connect the earthed electrode and  
the neuron output value output line VO1. This neuron  
output value output line VO1 is connected through a  
20 resistor in the load circuit with the power source  
VM01. As a result, if the amplification of the data  
line potential ~~(is ended)~~<sup>ends</sup> in the state started by the ✓  
arithmetic circuit starting signal  $\Phi_N$ , the potential  
of the neuron output value output line VO1 is dropped  
25 by a voltage proportional to the number of data lines  
which are raised to the high potential, i.e., Vcc.  
According to the present embodiment, therefore, the  
neuron output values can be expressed in terms of the  
potential drop of the neuron output value output line  
30 VO1. Incidentally, the provision of similar circuits  
at one side of the data lines  $DA11$ , - - -, and  $DA1p$  is  
to avoid the unbalance of the data line capacities for  
the same reasoning as that of the multiplier MT of  
Fig. 11(a). According to the embodiment thus far de-  
35 scribed, the neuron output values or the connection

weights expressed by the plural<sup>ity of</sup> memory cells can be read out to the neuron output value output line or the connection weight output line.

Fig. 18(c) shows one embodiment of the block BLK2 for calculating the product sum of the neuron output values and the connection weights and the nonlinear transfer function circuit D10. In Fig. 18(c), the block BLK2 is composed of the load circuit LD102 and the multiplier MT10. The neuron output value output lines VO1, VO2, - - -, and VOn and the connection weight output lines TO1, TO2, - - -, and TOn are connected with the gates of the MOS transistors M16c1 and M16c2 of the MT10, and the MOS transistors are connected in parallel with the MOS transistor M16c3, which has its gate fed with the arithmetic circuit starting signal  $\Phi_N$ , to connect the earthed electrode and the product/sum output line NO. On the other hand, the product/sum output line NO is connected through the resistor RO2 in the load circuit LD102 with the power source VM02. In the state in which the arithmetic circuit starting signal  $\Phi_N$  is at the high level, so that the present circuit is started, the potential of the product/sum output line NO is dropped the more for the larger sum of the products of the potentials of the corresponding neuron output value output lines VO1, VO2, - - -, and VOn and the connection weight output lines TO1, TO2, - - -, and TOn. As has been described hereinbefore, the potentials of the neuron output value output lines VO1, VO2, - - -, and VOn and the connection weight output lines TO1, TO2 and TOn are dropped substantially in proportion to the magnitudes of the neuron output values and the connection weights so that the potential of the product/sum output line NO becomes the higher for the larger product sum of the neuron output values and the connec-



tion weights. The product/sum output line NO is input<sup>ted</sup> to the nonlinear transfer function circuit D10. ✓  
The nonlinear transfer function circuit D10 can be constructed by connecting <sup>an</sup> n number of circuits ✓  
5 shown in Fig. 18(d) in parallel. The circuit of Fig. 18(d) is made like the nonlinear transfer function circuit D of Fig. 14(a) by combining the differential amplifier and the inverter. Since, however, the polarities of the product/sum output line NO and the  
10 product sum of the neuron output values and the connection weights are different between the embodiments of Fig. 11 and Figs. 18(a), 18(b) and 18(c), the resistor Rx of the differential amplifier of Fig. 18(d) is connected in the <sup>opposite</sup> position <sup>opposed</sup> to <sup>that of the</sup> ✓  
15 resistor R72 of Fig. 14(a). In Fig. 18(d), therefore, the output NVx <sup>transits</sup> <sup>turns</sup> to the high potential if the product/sum output line NO exceeds the reference voltage VRx (x = 1, 2, - - -, and p). If <sup>a</sup> p number of ✓  
such nonlinear transfer function circuits DSx are  
20 provided and if the reference voltage VRx is changed, as shown in Fig. 18(e), the change in the product/sum output line NO can be indicated by the number <sup>of</sup> such ✓  
ones of the p outputs NVx <sup>as</sup> <sup>that</sup> take the high potential. ✓  
According to the present invention, the characteristics of the nonlinear transfer function circuit can be  
25 easily varied by varying the value of the reference voltage VRx. Incidentally, <sup>the</sup> <sup>in which</sup> case the circuit shown ✓  
in Fig. 18(c) is used as the multiplier MT10, the potential variation of the product/sum output line NO  
30 is generally kept away from linearity for the magnitudes of the product sum of the neuron output values and the connection weights by the characteristics of the MOS transistors. It is, therefore, advisable to set the value of the reference voltage VRx by con-  
35 sidering the characteristics of the multiplier or the

adder so that the characteristics of the nonlinear transfer function circuit may take a desired shape. As the case may be, the characteristics of the individual chips may be made difficult to understand accurately because of the fluctuations of the production conditions. In this case, the known neuron output values and connection weights are actually written in the arrays A and B, and the potential of the product/sum output line NO in the arithmetic mode is measured so that the value of the reference voltage VRx may be resultantly trimmed to the desired characteristics.

Incidentally, [here will be omitted] the detail of the input/output circuit DIO10 of Fig. 18(a). The circuit <sup>for</sup> ~~of~~ reading or writing a plurality of memory cells in parallel can be easily realized by using a plurality of circuits which are similar to the read circuit OUT or the write circuit WR shown in Figs. 14(c), 14(e) and 14(f). Moreover, the structure of the clock generator 16 will be omitted but can be easily realized like the circuit used in the ordinary memory.

Next, the method for realizing the embodiment of Fig. 10, in which the neuron output values and the connection weights are binarily expressed in <sup>by</sup> a plural bits, will be described in connection with the embodiment of Fig. 18(a). In order to add the data expressed binarily with <sup>by</sup> a plural bits, as shown in Fig. 10(a), it is necessary to 'weigh' and add the data of the plural memory cells bit by bit. For this <sup>to take place</sup> necessity,

the potential of the neuron output value output line VO1 drops in proportion to the magnitude of the binary neuron output values, if the ratios of the gate width of the MOS transistors connected with the data lines in the voltage/current converters VI1, VI2, - - -, and VIp in Fig. 18(b) are 1 : 2 : 4 : , - - -, and :  $2^p$ .

If, therefore, similar circuits are used for other neuron output values or connection weights, the weighing additions can be realized, as shown in Fig. 10(a). The block BLK2 shown in Fig. 18(c) can be used <sup>because</sup> [as] it is [as] the multiplier. The nonlinear transfer function circuit has to be given the function of the AD converter for rewriting the arithmetic result output<sup>ted</sup> to the product/sum output line NO in the plural<sup>ty of</sup> memory cells in ~~the~~ binary notations. For this <sup>to take place</sup> [necessity], it is possible to use the embodiment shown in Fig. 19(a). In the embodiment of Fig. 19(a), a  $z$  ( $z = 2^p$ ) number of nonlinear transfer function circuits DS1, DS2, - - -, and DSz and an encoder are combined. These nonlinear transfer function circuits DS1, DS2, - - -, and DSz are given the characteristics shown in Fig. 19(b) by adjusting the reference voltage VRx with the circuit of Fig. 18(d). Then, the magnitude of the product sum of the neuron output values and the connection weights can be known like the embodiment of Fig. 18(c) from the number of ones of the outputs NA1, NA2, - - -, and NAz <sup>which</sup> ~~as~~ have the high potential. Then, the equivalent expressions of  $z$  bits have to be changed into binary expressions of  $p$  bits by the encoder so that <sup>they</sup> they have to be transmitted to the write circuit through the  $p$  number of output lines NV1, NV2, - - -, and NVp. It follows that the encoder of Fig. 19(a) may be given the input/output relations<sup>hip</sup> shown in Fig. 19(c). This encoder can be realized without difficulty. An example of the structure for  $p = 3$  is shown in Fig. 19(d). The present embodiment can be easily extended to the cases other than that for  $p = 3$ .

The description <sup>thus far</sup> [thus far] made<sup>d</sup> is exemplified by the multi-layered neural network. Despite [of] this <sup>ex-</sup>emplification, however, the present invention should

not be limited to the multi-layered neural network but can be applied to other types of networks by using the embodiments thus far described. Figs. 20(a) and 20(b) and Figs. 21(a) and 21(b) show embodiments for realizing the data processing using the Hopfield network according to the algorithm of Fig. 5(b). Fig. 20(a) shows an embodiment in which the unsynchronized Hopfield network is realized by using memory cells one by one for expressing the neuron output values and the connection weights. As has been described with reference to Figs. 2 and 3, the basic arithmetic method is commonly shared between the multi-layered network and the Hopfield network. In the Hopfield network, however, the <sup>calculations</sup> arithmetics are carried out by using the neuron output values from all the neurons including those of itself. In Fig. 20(a), therefore, all the neuron output values are stored in one word line of ~~the~~ array A. In ~~the~~ array B, as shown, the connection weights necessary for calculating one neuron output value are stored on a common word line. The updating of the neuron output values can be executed in the following manner. In order to update the neuron output value  $V_1$ , for example, the word line WA of ~~the~~ array A and the word line of  $j = 1$  of ~~the~~ array B are raised. As a result, the new neuron output value of  $g(T_{11}V_1 + T_{12}V_2 + \dots + T_{1n}V_n)$  is calculated. This value may be written in the memory cell which is located in the position of  $i = 1$  on the word line WA of ~~the~~ array A. The <sup>updated</sup> ~~updatings~~ of the other neuron output values are similar. The value  $V_4$ , for example, is updated by raising the word line WA of ~~the~~ array A and the word line of  $j = 4$  of ~~the~~ array B. As a result, the new value  $V_4$  of  $g(T_{41}V_1 + T_{42}V_2 + \dots + T_{4n}V_n)$  is calculated. This value may be written in the memory cell in the position of  $i = 4$  on the word line

WA of [the] array A. Thus, the [arithmetics]<sup>calculations</sup> of the un-  
synchronized Hopfield network can be executed by up-  
dating the neuron output values  $V_i$  in the desired or-  
der. The [arithmetics]<sup>calculations</sup> of the synchronized Hopfield  
5 network can be easily realized by using the memory  
cells on the word line WA1 of [the] array A for storing  
the neuron output values at present and by using the  
memory cells on the word line WA2 for storing the new  
neuron output values, as shown in Fig. 20(b). First  
10 [of all], the word line WA1 of [the] array A and the word  
line of  $j = 1$  of [the] array B are raised. As a result,  
the new value  $V_1$  of  $g(T_{11}V_1 + T_{12}V_2 + \dots + T_{1n}V_n)$   
is calculated. This value may be written in the  
memory cell in the position of  $i = 1$  on the word line  
15 WA2 of the array A. Subsequently, the neuron output  
values  $V_2, V_3, \dots$ , and  $V_n$  are updated and written  
in the memory cells on the word line WA2 of [the] array  
A. When [the] <sup>have been updated</sup> ~~updatings of~~ all the neuron output values  
[are ended], the <sup>process</sup> ~~updating of~~ the neuron output values is  
20 continued by interchanging the roles of [the] word lines  
WA1 and WA2 of [the] array A such that [the] word line WA2  
is selected for calculating the neuron output values  
[whereas the] <sup>and</sup> ~~word line~~ WA1 is selected for storing the  
neuron output values. From now on, the processings  
25 are likewise <sup>process</sup> ~~proceed~~ed by interchanging the roles of  
[the] word lines WA1 and WA2 of [the] array A. Thus, ac-  
cording to the embodiment of Fig. 20(b), the [arith-  
metics]<sup>calculations</sup> of the synchronized Hopfield network can be ex-  
ecuted.

30 Likewise, the Hopfield network can be realized by  
using a plurality of memory cells for expressing the  
neuron output values and the connection weights. Fig.  
21(a) shows an embodiment for realizing the unsyn-  
chronized Hopfield network by using [the]  $p$  and  $q$  num-  
35 bers of memory cells equivalently for expressing the

neuron output values and the connection weights. Like Fig. 20(a), all the neuron output values are stored in one word line of ~~the~~ array A. Here, ~~the~~ p number of cells express one neuron output value. ~~The~~ storage of the array B is <sup>stored</sup> made such that the connection weights necessary for calculating one neuron output value are arrayed on a common word line for every q number of cells. The ~~updating~~ of the neuron output values may <sup>updated as in</sup> be executed like the embodiment of Fig. 20(a). Since, however, ~~the~~ p number of memory cells are <sup>used</sup> individually ~~used~~ for expressing the neuron output values, a p number of output lines of the nonlinear transfer function circuit D are provided so that the ~~arithmetic~~ <sup>calculation</sup> results may be written in parallel in ~~the~~ p number of cells. The synchronized Hopfield network can also be easily realized like Fig. 21(b) if two word lines of ~~the~~ array A are used <sup>as in</sup> like Fig. 20(b). Likewise, it is quite natural that the synchronized and unsynchronized Hopfield networks can be realized by binary expressions using p and q numbers of memory cells for expressing the neuron output values and the connection weights, as shown in Fig. 10(a).

Figs. 8(a) and Figs. 20(a) and 20(b), and Fig. 9(a) and Figs. 21(a) and 21(b) present basically identical structures. If, therefore, the embodiments of Figs. 11 to 19, <sup>are used</sup> the data processings according to the embodiments of Figs. 20(a) and 20(b) and Figs. 21(a) and 21(b) can be easily realized. Incidentally, in the Hopfield network, the procedure of continuing the updating of the neuron output values falls in the so-called "local minimum", in which the energy is not the minimum, but ~~the~~ minimal, so that the neuron output values are not <sup>longer</sup> varied ~~any more~~. In order to avoid this, the well-known "quasi-annealing method" can be used. The method of changing the shape of the non-

linear transfer function gradually is known for realizing the quasi-annealing method, as<sup>2</sup> described on pp. 122 of Neural Network Processing (published by Sangyo Tosho and edited by Hideki Asou). According to the present invention, this method can be easily realized by switching a plurality of nonlinear transfer function circuits D having different characteristics and by controlling the characteristics of the nonlinear transfer function circuits D<sup>externally</sup> [from the outside].

Although [there have been described] examples in which the neuron output values and the connection weights have been handled as positive numbers mainly in the multi-layered or Hopfield network<sup>have been described</sup>, it may be convenient depending upon the application that<sup>either</sup> both or one of the two values can take positive and negative values. The present invention can be easily applied to such<sup>a</sup> case. Fig. 22 shows one embodiment of the present invention, in which both the neuron output values and the connection weights are enabled to take positive and negative values. In Fig. 22, the neuron output values are stored in [the] memory cell array A, and the connection weights are stored in [the] memory cell array B. The individual values are expressed with p or q bits, indicating the absolute magnitudes, and with 1 bit indicating the codes. The bits indicating the codes (as will be called the "coding bits") indicate a positive value with "1" and a negative value with "0". Of the neuron output values and connection weights, thus read out by the method similar to those described<sup>earlier</sup> [herein before], the portions of the p or q bits indicating the absolute values are input[ed] to the adders a1, - - -, and an and b1, - - -, and bn so that the resultant analog values are input[ed] to the multipliers m1, - - -, and mn. Incidentally, when the neuron output values and the connec-

tion weights are to be expressed in the binary notation, the data of the individual p and q bits inputted to the aforementioned adders  $a_1, \dots, a_n$ , and  $b_1, \dots, b_n$  may be weighed and inputted like Fig.

10. On the other hand, the coding bits are inputted as shown in Fig. 22, to exclusive OR circuits  $EOR_1, \dots, EOR_n$ . In <sup>the</sup> <sup>in which</sup> case the coding bits fail to become similar, namely, when the result of <sup>a</sup> multiplications is negative, the outputs of the aforementioned exclusive OR circuits <sup>become</sup> take the high level. In case of the similarity, namely, when the multiplication result is positive, the outputs of the exclusive OR circuits <sup>become</sup> take the low level. Switches  $SW_1, \dots, SW_n$  operate to transfer the outputs of the multipliers to the adders  $c_1, \dots, c_n$ , when the outputs of the exclusive OR circuits <sup>become</sup> take the low level, and the same or to the multipliers  $c'_1, \dots, c'_n$  when the outputs of the exclusive OR circuits <sup>become</sup> take the high level. As a result, the sum of the positive results of multiplications is outputted to the product/sum output line NO, and the sum of the negative multiplication results is outputted to the product/sum output line NO'. In the nonlinear transfer function circuit D, the difference between the signals of the product/sum output line NO and the product/sum output line NO' is converted into a digital value of p bits and fed to the bus ABS so that the coding bits are determined according to the magnitudes of the signals of the product/sum output line NO and the product/sum output line NO' and outputted to a bus SIGN. Incidentally, it is <sup>quite</sup> easily possible, according to the method similar to the aforementioned ones <sup>mentioned above</sup> to give the nonlinear characteristics, as shown in Fig. 9 or 10, according to the expressions of the neuron output values. According to the present embodiment, both the neuron output values



and the connection weights can take [the] positive and negative values. As a result, the present embodiment is advantageous in that the range for applying [the] data processing is extended. Although both the neuron output values and the connection weights are [enabled] to take [the] positive and negative values, it is easily possible to make modifications [such] <sup>so</sup> that either <sup>one</sup> of them take [the] positive values.

The description thus far made has been directed to the embodiment in which the product/sum function necessary for calculating the neuron output values and the nonlinear transfer function circuit are realized [as] <sup>in</sup> the arithmetic circuit. Despite [of] this description, however, a circuit for other [arithmetics] <sup>calculations</sup> can be added to the arithmetic circuit. For example, the data processing system according to the present invention can be applied to the so-called "classification problem" such as the speech recognition or the letter recognition, in which input patterns are classified into several classes. In this case, the comparator is conveniently disposed in the arithmetic circuit, as has been described <sup>earlier</sup> [hereinbefore]. In the classification problem, a desired value corresponding to a certain class can be attained as the output in <sup>the</sup> case <sup>where</sup> the inputted patterns are apparently classified <sup>into</sup> [to] the class. In the delicate case, however, in which it is questionable to determine which of <sup>the</sup> plural <sup>by</sup> classes the inputted pattern belongs to, the classification may fall <sup>at</sup> <sup>in</sup> a middle, between the desired values of the plural <sup>by</sup> classes. In <sup>the</sup> case <sup>where</sup> the inputted speech is 'K' in the speech recognition, for example, it is coded, and the connection weight is [so] set <sup>so</sup> that the neuron output value (or the desired value) of 1111 may be obtained in the output layer [for] <sup>of</sup> the speech waveforms [given to] the input layer. If the input is 'C', the connection

weight is [so] set<sup>so</sup> that the output value (or the desired value) of 0000 may<sup>result</sup> be issued. If, in this case, a middle speech waveform between [the] 'K' and [the] 'C',<sup>results</sup> the neuron output value of the output layer may<sup>give</sup> [output] a middle value such as 0001 or 1110. In this case, the distance (or similarity) between the neuron output value of the output layer and the desired value of 1111 for the 'K' or the desired value 0000 for the 'C' can be interpreted as measures [for giving]<sup>of</sup> the similarity to the 'K' or 'C' of the input speech. It is, therefore, convenient to give a function to determine the distance between the output result and the desired value by providing the arithmetic circuit with a circuit for comparing the neuron output value of the output layer and the desired value of the class.

Fig. 23 shows one embodiment [in] which [there are] integrated [in] one semiconductor chip: an arithmetic circuit 12a for comparing the neuron output values and the desired value; and an arithmetic circuit 12b for calculating the neuron output values. In Fig. 23: the desired value is stored in the memory TG; the neuron output values are stored in [the] memory A; and the connection weights are stored in [the] memory B. The calculations of the neuron output values may be accomplished<sup>realized</sup> by the method similar to those thus far described, by reading [out] the neuron output values from [the] memory A and the connection weights from [the] memory B, by calculating the neuron output values with the arithmetic circuit 12b and by writing the calculated result in [the] memory A. The comparisons are carried out by reading the neuron output values from [the] memory A and the desired value from [the] memory TG, by determining the distances in parallel with the arithmetic circuit 12B, and by writing the result in [the] memory TG or [outputting]<sup>sending</sup> the same through the

input/output device. Since, in the present embodiment, both [the] memories TG and A and the arithmetic circuit 12a are formed [over]<sup>on</sup> the common chip, the numbers<sup>be</sup> of [the] buses 1 and 2 can [be] easily increased to process the numerous bits in parallel. This results in [an advantage]<sup>the benefit</sup> that the distances can be calculated at [a] high speed. Incidentally, in the structure<sup>so</sup> [thus] far described, it is convenient to divide the arithmetic mode into<sup>a mode called</sup> the neuron output value calculating mode, [for]<sup>which</sup> calculating<sup>es</sup> the neuron output values and<sup>a mode called</sup> the comparison mode [for]<sup>which</sup> comparing<sup>es</sup> the neuron output values and the desired value to determine the distance. The switching of the arithmetic mode<sup>for example</sup> can be accomplished in response to the two arithmetic circuit control signals NE1 and NE2, [for example]. Specifically: the memory mode may be selected if both the signals NE1 and NE2 are [at the] high [level]; the neuron output value calculating mode may be selected if the signal NE1 is [at the] low [level] whereas the signal NE2 is [at the] high [level]; and the comparison mode may be selected if the signal NE1 is [at the] high [level] whereas the signal NE2 is [at the] low [level]. Incidentally, in the embodiment of Fig. 23, the memory is divided into [three]<sup>risks</sup> whereas the arithmetic circuit is divided into [two]<sup>halves</sup>, but these divided circuits may naturally be mixed [over]<sup>on</sup> the chip. As has been described<sup>above</sup> hereinbefore, according to the present embodiment, it is possible to determine<sup>at high speed</sup> the distances between the neuron output values and the desired value [at a high speed]. As a result, the data processing speed can be accelerated<sup>when</sup> [in case it is] necessary, as in [the] pattern recognition<sup>which</sup> using<sup>uses</sup> the multi-layered network [ ] to compare the neuron output values and each desired value to determine the distances [in]<sup>then</sup> between.

Fig. 24 shows one embodiment of the arithmetic

circuit 12a <sup>in</sup> of Fig. 23, i.e., a circuit for comparing the neuron output values of the output layer and the desired value to calculate the <sup>Hamming</sup> humming distances between. In the following, it is assumed that the memories TG and A of Fig. 23 are of the type, in which the data of the memory cells are read out to the data line pairs, as in Fig. 11, Fig. 16 or Fig. 18, and that the memories have the arrays TG and A, respectively. The circuit of Fig. 24 is composed of a comparator CMP and a comparison result converter COMPOUT. The comparator CMP is composed of a comparator CPU and a load resistor  $R_{CMP}$  connected in parallel, and the comparison result converter COMPOUT is composed of differential amplifiers AMP211, AMP212, - - -, and AMP21Z. The comparator CMP is connected with the data lines DTG1 and  $\overline{DTG1}$ , - - -, and DTGr and  $\overline{DTGr}$  of the array TG, and the data lines DA1 and  $\overline{DA1}$ , - - -, and DAR and  $\overline{DAR}$  of the array A. Here, letter r designates the number of memory cells on one word line and takes the value of n, if the neuron output values are expressed in 1 bit, and the product of n and p if the neuron output values are expressed in p bits. According to the present embodiment, it is possible to calculate the <sup>Hamming</sup> humming distances between the data read out onto the data lines DTG1 and  $\overline{DTG1}$ , - - -, and DTGr and  $\overline{DTGr}$  of the array TG and the data read out onto the data lines DA1 and  $\overline{DA1}$ , - - -, and DAR and  $\overline{DAR}$  of the array A. The operations of the present embodiment will be described <sup>below</sup> in the following. First of all, a clear signal  $\Phi_c$  is raised in advance to turn on a MOS transistor Q216 and to break the gate voltage of a MOS transistor Q215. After the clear signal  $\Phi_c$  has been broken so that the signal is read out to the data line to set the data line potential to  $V_{cc}$  or 0 V, the comparator is started by a comparator starting signal  $\Phi$

CMP. Then, the logic of exclusive OR is taken in each of the groups of the data lines (DTG1, DA1), (DTG2, DA2), - - -, and (DTGr, DAR) connected with the comparator. As a result, the gate of the MOS transistor Q215 is left at the low potential, in case <sup>where</sup> the data are similar on the data lines <sup>in</sup> at the array TG and the data lines at the array A, <sup>The gate of the MOS transistor Q215 is left low</sup> but otherwise <sup>is</sup> transited to the high potential. Then, the MOS transistor Q215 is turned on in the comparator CPU in which <sup>when</sup> the data are not <sup>the same</sup> similar between the data lines of the array TG and the data lines of the array A. As a result, the more current will flow from the power source VCMP through the load resistor RCMP to the earthed electrode for the larger number of groups of the data lines (DTG1, DA1), (DTG2, DA2), - - -, and (DTGr, DAR), in which the data are not similar. As a result, the potential of the compare line CO will be the lower for the larger number of groups in which the data are not similar. The compare line CO is connected with the differential amplifiers AMP211, AMP212, - - -, and AMP21Z, disposed in the comparison result converter COMOUT. If the reference voltages VRC1, VRC2, - - -, and VRCZ of those differential amplifiers are set to suitable values, the number of those of the comparison result output lines DCO1, DCO2, - - -, and DCOZ, which <sup>are</sup> take the high potential, is the larger for the larger drop of the potential <sup>on</sup> of the compare line CO. In other words, the comparison result converter COMOUT operates as a kind of AD converter. Thus, according to the embodiment of Fig. 24, the data read out to the plural <sup>by of</sup> data lines of the array TG and the data read out to the plural <sup>by of</sup> data of the array A can be compared to determine their <sup>bit of</sup> humming distance. If, therefore, one word is selected from each of the array TG and the array A, the data stored in the memory cells on the

selected word lines can be compared with each other. If, therefore, the desired values are individually stored in the memory cells on [the] array TG, it can be compared with the neuron output values [ ] which are stored in the memory cells on one word line of [the] array A<sub>6</sub>. <sup>This is done in order</sup> to know what desired value the neuron output values are close to and how close they are. <sup>Therefore, the</sup> <sup>where</sup> in case [ ] therefore, the obtained neuron output values are not similar to the desired value corresponding to the class, it is possible to know at [a] high speed what class the neuron output values are close to and how close they are.

Incidentally, in the embodiment of Fig. 24, the result outputted to the comparison result output lines may be outputted <sup>external to</sup> to the outside of the chip through the input/output circuit at each time of comparison. Alternatively, the capacity of [the] memory TG may be made larger than that necessary for storing the desired values so that the results may be <sup>at once</sup> once written in [the] memory TG and then outputted altogether.

Finally, an embodiment for further speeding up the system of the present invention by using a register will be described [in the following]. As has been described <sup>earlier</sup> hereinbefore, according to the present invention, the neuron output values are calculated by reading out the necessary data from the memory, by determining the neuron output values with the arithmetic circuit, and by rewriting the determined result in the memory. In other words, one arithmetic mode (i.e., the neuron output value arithmetic mode) cycle is composed of the reading operation and the writing operation, and the arithmetic circuit is inactive in the writing operation. If, therefore, the time period [ ] for which the arithmetic circuit is in [active] is shortened, <sup>speed of the</sup> the arithmetic mode can be further

<sup>increased</sup>  
speeded up<sup>^</sup>. Fig. 25 shows one embodiment in which the <sup>increased</sup> speed of the arithmetic mode is speeded up<sup>^</sup> on the basis of the  
aforementioned point of view. The embodiment of Fig. 25 is made by adding the register and the switches  
5 SW1, - - -, and SWr to the embodiment of Fig. 7. According to the embodiment of Fig. 25, the neuron output values can be calculated at [a] high speed<sup>^</sup> by using the algorithm of Fig. 5. In the following, the description will be made upon the multi-layered network,  
10 but similar effects can be attained even in the Hopfield network. In the embodiment of Fig. 25, the output value of the first neuron of the s-th layer is calculated by raising one word line of the memory cell array A to read out the neuron output value of the (s-1)th layer, by closing the switches SW1, - - -, and  
15 SWr to write the neuron output values of the (s-1)th layer in the register 14, and by opening the switches SW1, - - -, and SWr. Next, one word line of the memory cell array B is raised to read the connection weight between the neuron of the (s-1)th layer and the  
20 first neuron of the s-th layer, and <sup>then</sup> the neuron output values of the (s-1)th layer are read out by [the] register 14 so that the output value of the first neuron of the s-th layer is calculated by the arithmetic circuit  
25 12. The calculated results are written in [the] memory cell array A. <sup>at the same time</sup> [Simultaneously with this], one word line of [the] memory cell array B is raised to read out the connection weight between the neurons of the (s-1)th layer and the second neuron of the s-th layer, and the  
30 neuron output values of the (s-1)th layer are read out by the register 14 so that the output value of the second neuron of the s-th layer is calculated by [the] arithmetic circuit 12. <sup>after this</sup> [From now on], the output values of the neurons of the s-th layer are likewise calculated.  
35 Next, the output values of the neurons of the

(s+1)th layer are calculated by raising one word line of [the] memory cell array A to read out the previously determined neuron output value of the s-th layer, and by closing the switches SW1, - - -, and SWr to write the neuron output values of the s-th layer in [the] register 14, and the subsequent calculations [are] proceeded like<sup>as</sup> before. As has been described hereinbefore, according to the present embodiment, the writing operation and the reading operation can be simultaneously carried<sup>out at high speed</sup> by providing [the] register 14 [so that they can be accomplished at a high speed].

The description thus far made is directed mainly to the method of calculating the neuron output values according to the present invention, and the necessary connection weights are assumed to [be] already<sup>have been</sup> given. Depending upon the subject, the necessary connection weights are easily given at the start or have to be determined by the so-called "learning". In the learning for the multi-layered network called [the] "back propagation", for example, several neuron output values (or test patterns) of the input layer can be prepared in advance to determine the connection weights so that the desired neuron output values may be obtained in the output layer for the test patterns. As described in Section 2 of Neural Network Data Processing (published by Sangyo Tosho and edited by Hideki Asou), moreover, [there is known] the learning algorithm for setting the connection weights such that the balanced state of the neuron output values can take the desired state even in the Hopfield network. This learning can be applied to the present invention by the following three methods. According to the first method, the learning is carried out by using an external computer, and the obtained connection weights are written in the data processing system according to



the present invention. This method is advantageous [in  
 that <sup>since</sup> the learning can be executed by the software so ✓  
 that the learning algorithm can be easily changed, but  
 \* is difficult to speed up the learning. According to ✓  
 5 the second method, the arithmetic circuit of the sys-  
 tem according to the present invention is given an  
 arithmetic function for the learning so that the  
 learning is executed on-chip. This method speeds up  
 the learning but may be difficult to integrate all the  
 10 circuits necessary for the learning over a common  
 chip. The third method is an intermediate one between  
 the first and second methods, and a portion of the  
 [arithmetics] <sup>calculations</sup> necessary for the learning is executed by ✓  
 the system of the present invention, whereas the re-  
 15 maining portion of the [arithmetics] <sup>calculations</sup> necessary for the  
 learning is executed by the external computer. This  
 method is advantageous in that it can accelerate the  
 learning speed more than the first method and that the  
 arithmetic circuit of the system of the present inven-  
 20 tion can be simply constructed. This third method  
 will be specifically described <sup>below</sup> in the following. In- ✓  
 cidentally, the learning method is exemplified by the  
 back propagation method in the multi-layered network.  
 In the back propagation method (as will be shortly re-  
 25 ferred to as the "BP" method), the connection weights  
 are updated according to the following formulas:

$$T_{i,j}^{s+1} = T_{i,j}^s + \varepsilon d_{j,s} V_{i,s-1} \quad \text{--- (1);}$$

$$d_{j,m} = (t_j' - V_{j,m}) g'(U_{j,m}) \quad \text{--- (2);}$$

and

$$30 \quad d_{j,s} = g'(U_{j,s}) \sum_i (T_{i,j}^{s+1} d_{i,s+1}) \quad \text{--- (3)}$$

(s = m-1, ---, and 2),

wherein:

$\varepsilon$  : a small positive number;

35  $t_j$  : a target of the neuron output value  $V_{j,m}$   
 of the final layer;

$g'$ : a derivative of the nonlinear transfer function  $g$ ; and

$U_{js}$ : a quantity before passage through the nonlinear transfer function circuit  $g$  in the  $j$ -th neuron of the  $s$ -th layer, as will be defined by the following formula:

$$U_{js} = \sum_i (T^{s-1}_{ji} V_{is-1} + \Theta_{js}) \quad (4).$$

The connection weights may be updated by determining the quantities to be updated from the above-specified formulas (1) to (4) for every input data for the learning and by using the sum of the updated quantities of all the input data for the learning. On the other hand, the updating may be carried out by adding the following term called the "inertia term" to the formula (1):

$$\mu \Delta T^{s-1}_{ji} \quad (5),$$

wherein:

$\mu$ : a small positive constant; and

$\Delta T^{s-1}_{ji}$ : a corrected quantity of the previous updating.

The updating is continued <sup>until</sup> the difference between the neuron output values of the last layer and the target values becomes sufficiently small.

The learning <sup>so far</sup> described can be executed by the embodiment shown in Fig. 23 and the external computer, as will be described <sup>below</sup> in the following. The description to be made is directed to the case in which the updating is executed by summing the updated values of all the input data, but similar operations are applied to the case in which the connection weights are updated for every input data. Incidentally, the description to be made is directed to the case of the three-layered network, but similar operations are applied to the case of a network having

three or more layers.

First [of all], the input data for all the learn-  
ings and their target values are written in [the] memo-  
ries A and TG, respectively. Next, [the] a random num-  
5 ber having a small absolute value is written as the  
initial value of the connection weights in [the] memory  
B. Moreover, the first input data are read out as the  
neuron output values of the first layer to the arith-  
metic circuit 12b, and the connection weights between  
10 the first and second layers are read out to the arith-  
metic circuit 12b by [the] memory B. These values are  
multiplied in parallel by the aforementioned method so  
that the neuron output values of the second layer are  
calculated and written in [the] memory A. Subsequently,  
15 the neuron output values of the third layer are calcu-  
lated and written in [the] memory A. The calculations  
thus far described are executed for all the [learning]<sup>learning</sup>  
input data to read out the neuron output values of the  
individual layers for the individual input data, the  
20 desired values for the individual input data, and the  
connection weights to the memory outside of the chip.  
Next, the quantities for updating the connection  
weights are calculated in the external computer, and  
the updated connection weights are written in [the]  
25 memory B of the system according to the present inven-  
tion. Incidentally, the term  $g'(U_{js})$  appearing in [the]  
formulas (2) and (3) may be either calculated from the  
value  $U_{js}$ <sup>which is</sup> input[ed] to the nonlinear transfer function  
circuit D<sub>2</sub>] when the neuron output value  $V_{js}$  is to be  
30 calculated in the system of the present invention, or  
calculated inversely from the value  $V_{js}$  by the exter-  
nal computer according to the following formula:  
$$g'(U_{js}) = g'(g^{-1}(V_{js})) - - - - - (6).$$
  
In order to add the inertia term of [the] formula (5),  
35 on the other hand, the corrected quantities of the

connection weights may be stored in the memory outside of the chip for every updating[s] so that they may be added to the newly determined corrected quantities in accordance with [the] formula (5).

5        The updatings thus far described can be repeated  
[to proceed]<sup>λ<sub>0</sub></sup> the learning<sup>may proceed</sup>. In order to know how the  
learning advances, the distances between the neuron  
output values of the last layer for the individual input  
data and their desired values can be used as [the]  
10        measures. These distances can be calculated at [a] high  
speed by using the embodiment of Fig. 24. As a re<sup>progress</sup>-  
sult, it is easily possible to confirm the [advance] of  
[the] learning while the learning is being accomplished.

15        As has been described hereinbefore, according to  
the present invention, the calculations of the neuron  
output values for the input data for the learning can  
be executed at [a] high speed in the system of the pre-  
sent invention. In the present invention, moreover,  
the memories composed of memory arrays are used in [the]  
20        memories TG, A and B so that all the input data, the  
desired values and the neuron output values of the  
preceding layer can be easily stored and so that the  
numerous bits can be read out in parallel by raising  
the word lines. As a result, the transfers of the  
25        data to the external memory can be executed altogether  
at [a] high speed. As a result, the learning can [be]  
proceed[ed] at [a] high speed.

30        If the capacities of the memories are made sufficiently large in the present embodiment, the number of  
neurons can be easily changed according to the application. If, in this case, the neuron number is  
drastically changed, the nonlinear transfer function  
circuit may have to have its dynamic range changed.  
35        [For this necessity]<sup>if this is true</sup>, it is possible to switch and use a  
plurality of nonlinear transfer function circuits hav-

ing different characteristics and the reference voltages of the amplifiers in the nonlinear transfer function circuit. In <sup>the</sup> case <sup>in which</sup> the neuron numbers are different for the layers in the multi-layered network, the nonlinear transfer function circuit may have to have its dynamic range changed for the layers. This <sup>requirement</sup> (necessity) can also be <sup>dealt</sup> (coped) with by the similar method.

Incidentally, the description thus far made is directed to the embodiment in which either the DRAM cell of the so-called "one transistor and one capacity" type and the SRAM cells shown in Figs. 16(a) and 16(b) are mainly used, but other memory cells can naturally be used in the present invention. Since the portion for storing the connection weights need not be frequently rewritten for ~~(the)~~ data processing, ~~the~~ kinds of ~~(the)~~ cells can be changed according to the contents of the memories by using non-volatile memory cells or the DRAM cells or the SRAM cells in the portion for storing the neuron output values.

If the memory cell circuit is highly integrated by using very small memory cells such as ~~(the)~~ DRAM cells of the one transistor and one capacitor, some memory cells may ~~(be)~~ sometimes <sup>be</sup> inactive because the wiring lines used are very small. The neural network is advantageous in that its function is hardly influenced even if the connection weights are changed more or less, but the data processing may be troubled <sup>the</sup> in <sup>in which</sup> case the memory cells for storing the neuron output values are inactive. In order to solve this problem, the redundant word lines or data lines to be used in the ordinary highly-integrated semiconductor memory can be provided so that defective cells may not be used.

In Figs. 14(a), 14(e) <sup>and</sup> 14(f) and Fig. 18(d),

moreover, [there are used the] bipolar transistors<sup>are used</sup>,  
which can be realized by the CMOS. Still moreover,  
the present invention can be practiced by not only the  
bipolar transistors and the MOS transistors but also<sup>by</sup>  
5 other devices.

Although the foregoing description is directed  
mainly to the multi-layered and Hopfield networks, the  
present invention should not be limited thereto but  
can be applied to [the] neural network data processing  
10 for [the] networks of various types. For example, it is  
possible to realize the network in which [the updating  
of] the neuron output values<sup>are updated</sup> such as<sup>in</sup> the Boltzman's  
machine. As described on pp. 27 of Neural Network  
Data Processing (published by Sangyo Tosho and edited  
15 by Hideki Asou),<sup>where</sup> the Boltzman's machine is featured,  
although the network shape is similar to that of the  
Hopfield network, in that the neuron output value (0  
or 1) is not uniquely determined by another product  
sum of the neuron output values inputted to the neu-  
20 rons and the connection weights, but in a probable man-  
ner. The probability P for the neuron output value<sup>to</sup>  
take the value 1 is expressed by  $P = 1 / (1 + \exp(-I/T))$ .  
Here, letter I designates the product sum of the  
not inputted to the neurons and the connection wei-  
25 ghts, and letter T designates a parameter called the  
temperature. The Boltzman's machine described above  
can be easily realized according to the present inven-  
tion. For example, the reference voltage VRx of the  
nonlinear transfer function circuit D, as shown in  
30 Fig. 18(d), is not set to [the]<sup>a</sup> steady value but may be  
changed with time within the fluctuating range of the  
product/sum output line NO. Then, the neuron output  
values can be determined according to the probability.  
The effects obtained by changing the changing rate are  
35 similar to those obtained by changing the temperature

T.

If the capacity of the memories is sufficient, as is apparent from the comparisons between Fig. 8(a) and Figs. 20(a) and 20(b), various types of network<sup>s</sup> can be realized by a common system merely by changing the addresses of the memory cells for storing the neuron output values and the connection weights. Thus, the present invention has a [highly]<sup>very</sup> wide applicability.

Although the description thus far made is directed to the applications ~~to~~<sup>of</sup> the neural network data processing, the present invention should not be limited thereto but can naturally realize such a system ~~in~~<sup>with</sup> a high degree of integration as<sup>t</sup> is used for the data processing by connecting a number of processing elements having similar processing functions in the form of a network.

In the embodiments thus far described, the description is directed mainly to the structure in which the arithmetic circuit performs the analog<sup>calculations</sup> arithmetic<sup>s</sup>. The analog arithmetic circuit is advantageous in that it has a high speed and a small circuit scale. Despite ~~of~~ this description, however, the present invention should not be limited thereto but can be used in a digital arithmetic circuit without departing from the gist thereof. In this case, the calculations can be [highly accurately]<sup>very accurately</sup> executed<sup>^</sup> by the digital arithmetic circuit.

As has been described hereinbefore, according to the present invention, the system for executing the data processing by combining the memories and the arithmetic circuits and by performing the parallel<sup>calculations</sup> arithmetics<sup>s</sup> with the arithmetic circuits like the parallel distributed processing system such as the neural network, in which a number of arithmetic units for relatively simple arithmetics<sup>calculations</sup> are connected in the

network form, can be realized with a high degree of integration without sacrificing [the] speed. ✓

5 It is further understood by those in the art that the foregoing description is <sup>the</sup> preferred embodiment of ✓ the disclosed device and that various changes and modifications may be made in the invention without departing from the spirit and scope thereof.



# ABSTRACT OF THE DISCLOSURE

Herein disclosed is a data processing system having a memory packaged therein for realizing [a] large-scale and high-speed parallel distributed processing and, especially, a data processing system for [the] neural network processing. The neural network processing system according to the present invention comprises: a memory circuit for storing neuron output values, connection weights, the desired values of outputs, and data necessary for learning; an input/output circuit for writing or reading data in or out of said memory circuit; a processing circuit for performing a processing [ ] for determining the neuron outputs such as the product, sum and nonlinear conversion of the data stored in said memory circuit, a comparison of the output value and its desired value, and a processing necessary for learning; and a control circuit for controlling the operations of said memory circuit, said input/output circuit and said processing circuit. The processing circuit is constructed to include at least one [of an] adder, a multiplier, a nonlinear transfer function circuit and a comparator so that at least a portion of the processing necessary for determining the neuron output values such as the product or sum may be accomplished in parallel. Moreover, these circuits are shared among a plurality of neurons and are operated in a time sharing manner to determine the plural<sup>ty</sup> of neuron output values. Still moreover, the aforementioned comparator compares the neuron output value determined and the desired value of the output in parallel.